Андрей Робачевский

Интернет изнутри

Архитектура экосистемы Интернета

indata
ФОНД РАЗВИТИЯ СЕТЕВЫХ ТЕХНОЛОГИЙ «ИНДАТА»

Моим женщинам:
маме, жене и дочери

Андрей Робачевский

ИНТЕРНЕТ ИЗНУТРИ

Архитектура экосистемы Интернета

3-е издание, переработанное и дополненное

ФОНД РАЗВИТИЯ СЕТЕВЫХ ТЕХНОЛОГИЙ «ИНДАТА»

Москва
2024

УДК 004.738.5
ББК 32.973.202
Р12

Редактор И. Пыжова

Робачевский А.

Р12 Интернет изнутри: Архитектура экосистемы Интернета / Андрей Робачевский. — 3-е изд., перераб. и дополн. — М.: Серпантин Эдженси, 2024. — 300 с.

ISBN 978-5-6052033-0-8

Книга рассказывает об архитектуре и технологиях Интернета, фокусируясь на его основных компонентах: глобальной адресации и протоколе IP, системе доменных имен и глобальной межсетевой маршрутизации. Рассматриваются аспекты и принципы работы Всемирной сети, вопросы стандартизации, развития и безопасности основных систем Интернета. Обсуждается архитектурная эволюция Интернета в целом, а также связанные с ней вопросы внедрения новых протоколов и технологий.

Особое внимание уделено экосистеме Интернета, ее истории, а также основным организациям, включенным в систему принятия решений в Интернете.

Книга рассчитана на техническую аудиторию: сетевых операторов (администраторов), разработчиков программного обеспечения. Она также будет полезна тем, кто интересуется архитектурными аспектами Сети, вопросами «управления» Интернетом, и всем желающим расширить свой кругозор в области интернет-технологий.

УДК 004.738.5
ББК 32.973.202

ISBN 978-5-6052033-0-8

Содержание

Предисловие

К первому изданию

Интернет — это просто и легко?

Казалось бы, чего проще — нажал кнопку и получил письмо (зашел на сайт, прочитал афишу, ленту новостей, обменялся комментариями, заказал билет). Мы делаем это ежечасно, ежеминутно. Интернет давно стал неотъемлемой частью нашей жизни уже настолько, что если где-то нет интернет-связи — это вызывает искреннее возмущение.

Миллиарды человек во всем мире используют Интернет, не задумываясь над тем, что за внешней простотой скрывается сложная инфраструктура.

Что происходит после нажатия на Enter? Куда и как идут данные? Почему возникают сбои в Сети? Как защититься в Сети? Почему все так сложно, если все так просто?

А что внутри?

Миллионы сетей в разных странах, находящихся под автономным управлением, умудряются работать четко и слаженно, как единый организм. Телекоммуникационная инфраструктура, по которой передаются потоки данных, технические стандарты и услуги, благодаря которым Интернет работает, стандарты контента и приложений, безопасность и стабильность — вот основополагающие вопросы, связанные с функционированием Интернета.

Андрей Робачевский написал замечательную книгу «Интернет изнутри», альтернатив которой вы не найдете сегодня. В ней автор дает доходчивые ответы на многие вопросы об устройстве Интернета. Профессионализм автора, который с начала 1990-х гг. работает в интернет-отрасли, и глубокое понимание основополагающих вопросов, связанных с функционированием Интернета, лежат в основе всех статей, составляющих эту книгу.

MSK-IX, начиная с 1995 г., поддерживает и развивает инфраструктурные проекты в сети Интернет. Мы выступили инициатором создания этой замечательной книги и подготовили ее издание специально к юбилейному, X Пиринговому форуму MSK-IX.

Я буду рада, если эта книга поможет всем нам еще на шаг приблизиться к пониманию устройства Интернета, к пониманию того,что за внешней простотой лежат сложные технологии и процессы, работа многих профессионалов и уникальных специалистов.

Елена Воронина,
исполнительный директор MSK-IX,
2014 г.

Ко второму изданию

Устройство Интернета — это просто

Вы держите в руках уже второе издание книги Андрея Робачевского «Интернет изнутри». Вышла эта книга в 2014 г., и ее сигнальный тираж был представлен на юбилейном, X Пиринговом форуме MSK-IX.

Мы выступили инициатором создания печатной версии, потому что хорошо понимали: эта тема — тема устройства и функционирования Интернета — интересует очень многих, и даже для профессионалов IT-индустрии может стать полезным содержание этой книги.

Однако мы не ожидали, что «Интернет изнутри» будет иметь такой успех! Мы выпустили журнал с одноименным названием, который начал выходить осенью 2015 г. — его главным редактором стал автор книги Андрей Робачевский. Наши читатели с удовольствием знакомились с номерами журнала — и продолжали спрашивать, когда же можно будет получить книгу.

Думаю, что секрет обаяния и успеха книги очень прост: автор сумел простым языком рассказать об очень сложных вещах.

За два года, прошедшие с выхода первого издания «Интернета изнутри», мир изменился, изменилась и книга. Вашему вниманию предлагается дополненное издание.

Книга «Интернет изнутри» уже стала бестселлером, и я уверена, что второе издание займет достойное место на книжных полках и столах наших друзей, коллег, партнеров и всех тех, кому интересен Интернет.

Желаю Вам приятного чтения и знакомства с удивительным миром ИНТЕРНЕТ!

Елена Воронина,
генеральный директор MSK-IX,
2016 г.

К третьему изданию

Интернет как технология, пожалуй, не самая популярная тема для публикаций в медиапространстве. Гораздо привлекательнее обсуждать его влияние на социальную жизнь современного общества и возможности, которые открылись благодаря использованию Интернета в различных сферах деятельности человечества. Эта технология поразительно изменила нашу жизнь за последние десятилетия: широкая доступность информации, возможность удаленной работы, обучения, медицинских консультаций, финансовых транзакций – этот список можно продолжать бесконечно.

Именно широкое проникновение интернет-технологий в повседневную жизнь общества налагает особые требования по управлению процессами в Интернете: обеспечение стабильности работы сети и безопасности ее компонентов, защита инфраструктуры, оптимальность маршрутов и пр.

Эта книга, созданная профессионалом с многолетним опытом работы в Интернете, призвана помочь сформировать цельный взгляд и понимание основных процессов управления всемирной сетью. Именно эту цель преследовал Фонд «Индата», инициировав появление на свет новой, уже третьей редакции книги «Интернет изнутри» и ее последующей публикации.

Издание этой книги органично вписывается в ряд инициатив Фонда, нацеленных на изучение и исследования Интернет, таких как журнал «Интернет изнутри», «Макроскопические исследования Интернет-инфраструктуры IDIDB.RU», образовательный проект и другие.

Публикация книги организована на сайте Фонда на безвозмездной основе и будет доступна всем желающим погрузиться в увлекательный мир внутреннего устройства Интернета. На этом же сайте опубликованы номера журнала «Интернет изнутри», каждый номер является тематическим и раскрывает подробности технологических аспектов сетевых технологий. Книга и журнал взаимно дополняют друг друга и являются частями научно-образовательной деятельности Фонда.

Хочу поблагодарить замечательного автора Андрея Робачевского за неизменную готовность к сотрудничеству, легкость изложения сложного материала и перфекционизм в деталях.

Надеюсь, что прочтение книги поможет вам осознать гармоничность и многоуровневость устройства всемирной сети. Присущую ей простоту в сложности и сложность в простоте как залог гениальности такого явления как Интернет.

Приятного и полезного чтения!

Елена Воронина,
директор по развитию Фонда развития сетевых технологий «Индата»

От автора

К первому изданию

Универсальный коннектор

В моем смартфоне 15 новых событий: у приятеля сегодня день рождения — не забыть поздравить его в «Фейсбуке»; знакомый пытался связаться по «Скайпу» и оставил видеосообщение; судя по фотографиям в «Инстаграме», которые опубликовал мой брат, в Санкт-Петербурге отличная погода. В почтовом ящике — новые письма, новостной сайт предлагает свежую подборку на интересующие меня темы, авиакомпания обнадеживает, что мой рейс задержится всего на 30 минут...

Типичный момент из жизни сотен миллионов людей во всем мире.

Интернет меняет наше представление о расстоянии и времени. Информация, личные данные и даже сами человеческие отношения приобретают новое измерение благодаря Сети. Интернет сегодня — нечто гораздо большее, чем технологии и протоколы, глобально взаимосвязанные сети и разнообразные устройства, онлайн-услуги, приложения и колоссальные объемы информации. Это — экосистема, живущая и развивающаяся по своим законам. Это — универсальный коннектор, поглощающий любой подключенный к нему элемент, который, в свою очередь, сам становится частью Сети.

Но чтобы лучше понять законы, по которым живет Интернет, нам придется разобраться, как работают его основные подсистемы и как они взаимодействуют между собой. Мы заглянем за облачный фасад приложений и услуг, чтобы увидеть Интернет изнутри. Мы узнаем, как зародились и развились протоколы, технологии и взаимоотношения, составляющие основу Всемирной паутины. К счастью, нам не придется совершать археологическое исследование, наш прыжок в прошлое будет длиной лишь одно поколение. Ведь уникальность интернет-революции еще и в том, что ее непосредственными участниками являемся мы сами.

А. Робачевский,
2014 г.

Ко второму изданию

С момента выхода в свет первого издания этой книги прошло два года, и меня по-прежнему удивляют динамика развития, инновационный потенциал и та роль, которую Интернет играет в нашей жизни.

Более заметны и новые тенденции.

Благодаря своему качеству универсального коннектора Интернет превращается в «гравитационную силу», когда различные объекты нашего физического мира не только могут, но и неизбежно должны быть подключены к этой глобальной системе. Мы становимся окружены компьютерами, которые мы по-прежнему называем холодильниками, телевизорами, осветительными лампами или автомобилями. И хотя они принадлежат нам, значительная часть их функциональности и ресурсов находится в Сети.

Сеть сама превратилась в глобальный виртуальный компьютер, обладающий неимоверными ресурсами, данными, информацией и знаниями. И возможностями. И в то же время сам термин «интернет» становится менее осязаем, имеющим различные значения для разных людей, в зависимости от контекста и конкретной темы. Истинную красоту этой системы мы сможем понять, только взглянув на нее изнутри.

А. Робачевский,
2016 г.

К третьему изданию

Прошло восемь лет с момента выхода в свет предыдущего издания книги. Восемь лет – это большой срок для такой динамичной экосистемы как Интернет. Увеличился географический охват (66,2% мирового населения сегодня используют Интернет по сравнению с 47% в 2016 году), существенно расширился спектр используемых технологий и особенно технологий защиты данных (согласно Google, процент веб-трафика, который использует шифрование, увеличился с 80% до 96%), а разнообразие приложений и способов использования Интернета поражают воображение (последний год прошел под знаком стремительного развития приложений генеративного искусственного интеллекта). И не менее замечательным является то, что архитектурные принципы и технологический фундамент Интернета не претерпели существенных изменений. Более того, они явились необходимым фактором и катализатором этого роста. Наша книга посвящена этой поразительной области киберпространства.

Итак, дорогой читатель, я приглашаю вас еще раз отправиться вместе со мной в это путешествие по постоянно меняющемуся ландшафту киберпространства. Давайте вместе погрузимся в глубины цифровой сферы, разгадывая ее тайны и раскрывая ее чудеса. И когда мы будем внимательно рассматривать Интернет изнутри, нам раскроется мир безграничных возможностей и непредсказуемых инноваций.

Я надеюсь, что «Интернет изнутри» предоставит всесторонний обзор прошлого, настоящего и будущего Интернета и послужит ценным ресурсом для студентов, ученых, практиков и энтузиастов.

А. Робачевский,
2024 г.


АДРЕСНОЕ ПРОСТРАНСТВО
ПОЛИТИКА РАСПРЕДЕЛЕНИЯ
RFC1518
IPv4
СОСУЩЕСТВОВАНИЕ
RFC4291
ВНЕДРЕНИЕ
АДРЕС
ВНЕДРЕНИЕ
УЗЕЛ
APNIC
RIPE NCC
TCP/IP
NAT64
ICANN
IPv6
ICANN
ПРОТОКОЛ
РАСШИРЕННЫЙ ЗАГОЛОВОК
RIPE-140
RFC1918
PA
ПОРТ
APNIC
RFC1518
RIPE-140
РЕГИСТРАТУРА
ИНТЕРНЕТ
ASO
/8
АДРЕС
TCP/IP
ЗАГОЛОВОК
ASO
RFC1518
NAT64
IPv4
ПОДКЛЮЧЕНИЕ
АДМИНИСТРИРОВАНИЕ
ИНТЕРНЕТ
ЗАРЕЗЕРВИРОВАНЫЕ АДРЕСА
АДМИНИСТРИРОВАНИЕ
RFC4291
СОСУЩЕСТВОВАНИЕ
ГЛОБАЛЬНАЯ ПОЛИТИКА
MSK-IX
БЕЗОПАСНОСТЬ
ДВОЙНОЙ СТЕК
TCP/IP
РИР
ФРАГМЕНТАЦИЯ
ПОЛИТИКА РАСПРЕДЕЛЕНИЯ
RIPE NCC
ВНЕДРЕНИЕ
RIPE NCC
DS-lite
СОСУЩЕСТВОВАНИЕ
ВНЕДРЕНИЕ
P1
RFC1174
АДРЕС
ПАКЕТ
RIPE NCC
RFC4291
ARIN
IPv4
ВНЕДРЕНИЕ
TCP/IP
CIDR
PA
AfriNIC
RFC4291
RFC1174
NRO
RFC1518
ICANN
RIPE-140
АДРЕСНЫЙ ПУЛ
AfriNIC
ЗАГОЛОВОК
RFC1174
RIPE NCC
УЗЕЛ
БЕЗОПАСНОСТЬ
646XLAT
IETF
RFC4291
/22
/32
LACNIC
RFC1174
RFC1518
RIPE-140
APNIC
/8
ЗАГОЛОВОК
ВНЕДРЕНИЕ
СОСУЩЕСТВОВАНИЕ
ICANN
СЕТЬ
RFC1174
DS-lite
RFC1174
APNIC
ИНТЕРНЕТ

Глава 1

Интернет-протокол IP и глобальная система адресации

> Следует определить различия между именами, адресами и маршрутами. Имя определяет то, что мы пытаемся найти. Адрес указывает, где это находится. Маршрут показывает, как туда попасть.
>
> *RFC 760[1], первая спецификация интернет-протокола IPv4, 1980 г.*

Три дня рождения Интернета

Ранним утром 29 октября 1969 года произошло историческое событие — рождение Интернета. В тот момент мало кто осознавал значимость этого события. Чарли Кляйн (Charley Kline) на своем терминале в Калифорнийском университете в Лос-Анджелесе (UCLA) набрал слово LOGIN, чтобы отправить эту команду компьютеру в Стэнфордском исследовательском институте (SRI), за которым ожидал коллега Чарли, Билл Дювал (Bill Duvall). Первый символ 'L' проделал путь в 500 км, был принят компьютером Билла и послан обратно, появившись на терминале Чарли. За ним последовал символ 'O'. На символе 'G' система сломалась, но была полностью восстановлена часом позже. Так был рожден Интернет.

Чарли и Билл были молодыми программистами, сотрудниками двух крупнейших американских научных центров. Рождению Интернета предшествовало десятилетие научных исследований, а свой вклад внесли десятки, если не сотни людей, разработавших базовые концепции архитектуры Интернета.

---

[1] RFC 760: DoD Standard Internet Protocol, URL: https://www.rfc-editor.org/rfc/rfc760

Еще в начале 1960-х гг. ряд исследователей, многие из которых в дальнейшем участвовали в проекте ARPANET, увидели огромные перспективы в способности компьютеров обмениваться друг с другом данными. В 1965 году было установлено тестовое соединение между компьютерами Массачусетского института технологии и Университета Южной Калифорнии — использовалась традиционная телефонная технология синхронной коммутации каналов. Стало очевидно, что такая коммутация не позволяет эффективно использовать канал связи, но именно в ходе этого эксперимента начал обретать очертания «эмбрион» будущего Интернета.

Слово «Интернет» вошло в обиход в середине 1970-х, а до того Сеть называлась ARPANET. По сравнению с телефонными сетями, основанными на коммутации каналов, в ARPANET было решено использовать технологию коммутации пакетов, или дейтаграмм — данных ограниченного объема, заключенных в «конверты» с указанием источника и получателя. Поскольку каждый пакет обрабатывался независимо, сети не требовалось хранить информацию о соединениях между оконечными компьютерами и потоках данных между ними. Этот подход позволил существенно упростить архитектуру сети и повысить ее надежность. Узел сети мог выйти из строя — но его функцию немедленно брал на себя другой, рабочий узел. Кроме того, асинхронная пакетная передача больше соответствовала характеру работы многозадачных операционных систем. Так, ОС Unix позволяла разделять ресурсы между несколькими задачами одновременно — процессор занимался и обработкой команд с многочисленных терминалов, и вычислением крупных массивов данных. Telnet (удаленный доступ в режиме терминала) и электронная почта (e-mail) появились в ARPANET в 1972 г., а ftp (обмен файлами) — годом позже. Первое время для обмена данными между компьютерами, или хостами, использовался протокол NCP (Network Control Protocol), предтеча сегодняшнего TCP/IP.

Функциональность протокола NCP ограничивалась тем, что это, по существу, был транспортный протокол. Он не был хорошо приспособлен для работы с разнообразными технологиями — например, цифровой радио- и спутниковой связью. Более того, он предназначался для работы только с одной сетью — ARPANET, а значит, не был способен осуществлять адресацию в других сетях и среди подключенных к ним компьютеров.

В это же время Роберт Кан (Robert Kahn), сотрудник агентства передовых исследовательских проектов DARPA, работал над концепцией открытой сетевой архитектуры. В рамках этой концепции независимые сети, различные по своей архитектуре и используемым технологиям, должны были свободно обмениваться данными. Требовалась лишь единая межсетевая модель для «прозрачного» обмена данными между компьютерами в различных сетях. Особенностью концепции Кана было то, что он рассматривал и функциональность беспроводных сетей пакетной коммутации. Поскольку радиосигнал может подвергаться искажениям до полной потери (например, при перемещении в туннеле), протокол должен был обеспечить надежную передачу данных независимо от качества сети.

В 1973 году Кан начал разработку протокола, который позволил бы передавать данные между хостами, используя любую коммуникационную технологию. Кан пригласил в свой проект Винтона Серфа (Vinton Cerf), в то время сотрудника Стэнфордского университета. Серф обладал необходимым опытом: ранее он участвовал в создании протокола NCP и разрабатывал сетевые интерфейсы к различным операционным системам. Благодаря совместным усилиям Кана и Серфа концепция нового протокола была представлена уже в сентябре 1973 года а годом позже, в декабре 1974-го, Серф вместе со своими аспирантами Йогеном Далалем (Yogen Dalal) и Карлом Саншайном (Carl Sunshine) опубликовал первую полную спецификацию протокола TCP. Аббревиатура означала Transmission Control Program, а сам протокол объединял в себе функции сегодняшних протоколов TCP и IP. Новейшая спецификация была зафиксирована в серии документов Request for Comments (RFC) под номером RFC 675.[2]

Интересно, что изначально архитектура протокола TCP предполагала использование 4 бит для адресации сети (допуская тем самым существование 16 сетей, из которых шесть были уже назначены: ARPANET, UCL, CYCLADES, NPL, CADC, EPSS), а также 16 бит для адресации хостов в сети (или «процессов TCP»). При этом заголовок пакета также содержал поле длины сетевого адреса, тем самым обеспечивая расширение адресного пространства при необходимости до 64 бит. Однако в следующей версии протокола TCP, опубликованной в 1977 году., уже использовались только адреса фиксированной длины. А ведь изначальная структура TCP предлагала более элегантный способ борьбы с нехваткой адресного пространства, нежели создание протокола IPv6, несовместимого с IPv4!

В том же 1977 году была проведена первая серьезная демонстрация работы «Интер-Нета»: три сети, использующие различные сетевые технологии — ARPANET, SATNET и сеть пакетного радио, — успешно обменивались данными по протоколу TCP. Так Интернет был рожден во второй раз.

Двумя месяцами позже в то время аспирант Калифорнийского университета в Лос-Анджелесе (UCLA) Джон Постел (Jon Postel) опубликовал статью, где предложил новый архитектурный взгляд на TCP — протокол, состоящий из двух компонентов. Первый компонент, который в последующей спецификации TCP получит название IP (Internet Protocol), отвечал только за передачу пакетов между узлами сети и маршрутизацию. Второй же компонент, TCP, обеспечивал сквозной поток данных между оконечными устройствами, контроль ошибок и повторную передачу потерянных данных.

В 1978 году Постел публикует четвертую версию протоколов IP и TCP. И наконец в 1980 году публикуется документ RFC 760[3], содержащий спецификацию IPv4 и принципы архитектуры Интернета, какими мы их знаем сегодня.

---

[2] RFC 675: Specification of Internet Transmission Control Program, URL: https://www.rfc-editor.org/rfc/rfc675

[3] RFC 760: DoD Standard Internet Protocol, URL: https://www.rfc-editor.org/rfc/rfc760

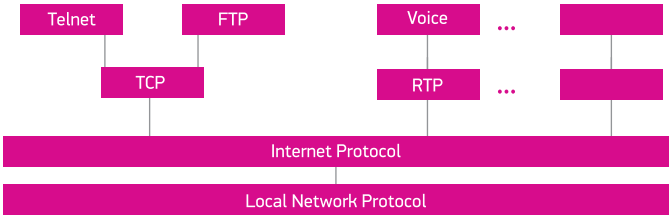

Рис. 1. Взаимодействие протоколов стека TCP/IP, определенное в спецификации RFC 760.

В рамках этой архитектуры IP отвечает за адресацию и фрагментацию пакетов при передаче от одного узла сети к другому. Адреса позволяют узлу принять решение, какому следующему узлу направить данные, а с помощью фрагментации данные можно передавать между сетями с различными допустимыми размерами пакета. Этим, собственно, функции протокола IP и ограничиваются.

В своей работе IP опирается на протоколы нижнего уровня, используемые в локальной сети, и транспортные протоколы, например, TCP. Сам же интернет-протокол не обеспечивает надежную передачу. В спецификации RFC 760 указано, что в протоколе IP «отсутствуют подтверждения, как сквозные, так и межузловые. Отсутствует контроль ошибок, за исключением контрольной суммы заголовка. Отсутствует функция повторной передачи. Отсутствует управление потоком данных».

Переход к семейству протоколов TCP/IP

Если вы думаете, что с переходом к протоколу IPv6 Интернет впервые переживает столь фундаментальное изменение базового протокола, то это не так. Сеть ARPANET конца 1970-х по-прежнему использовала протокол NCP, ограничивая возможности прозрачного обмена данными с другими сетями, например, с сетями пакетного радио или спутниковыми сетями. А в этом и заключалась основа концепции Интернета.

В ноябре 1981 года Джон Постел опубликовал план перехода ARPANET от протокола NCP к протоколам TCP/IP[4]. Учитывая, что новый протокол уже прошел успешное тестирование в различных конфигурациях, на переход отводился один год.

Тогдашнюю ARPANET невозможно сравнить с сегодняшней сетью Интернет — и по размеру, и по зависимости общества и экономики от ее функционирования, и по степени контроля и координации. Тем не менее, переход занял целый год

[4] RFC 801: NCP/TCP transition plan, URL: https://www.rfc-editor.org/rfc/rfc801

и потребовал определенного количества напоминаний и увещеваний со стороны Джона Постела. Кроме того, на сутки был отключен протокол NCP по всему ARPANET'у, так что только узлы, поддерживавшие протокол TCP/IP, могли обмениваться данными.

Окончательный переход на TCP/IP произошел, как и было запланировано, 1 января 1983 года Так Интернет был рожден в третий раз, теперь с протоколом IPv4.

Эволюция системы адресации: от протокола IPv4 к протоколу IPv6

В 1981 году трудно было представить, что 32 бита адреса IPv4, позволяющие присвоить уникальный номер четырем миллиардам систем (компьютеров, маршрутизаторов и т.п.), когда-либо станут реальным ограничением. Однако уже к 1992 году масштабируемость и ограниченность адресного пространства IPv4 встала на повестку дня.

Для поиска решения проблемы в ноябре 1991 года организация по стандартизации IETF сформировала специальную группу для «мозгового штурма» в области маршрутизации и адресации — ROAD (Routing and Addressing). Учеными было найдено краткосрочное решение проблемы: они предложили супернеты — концепцию, впоследствии переработанную в архитектуру CIDR (Classless Inter-Domain Routing, бесклассовая междоменная маршрутизация). Этот подход, который был стандартизован в 1993 году (RFC 1518[5], RFC 1519[6]), позволил существенно замедлить расходование запаса доступных адресов. В чем заключалась суть концепции CIDR? Граница подсетей становилась подвижной в зависимости от фактического размера адресуемой сети. Вместо распределения сетей класса C (/24) фиксированного размера (254 устройства) стало возможным создавать сети /23, /22 и так далее. Внутри же сервис-провайдер мог создать структуру, более соответствующую реальной топологии, распределяя сети меньшего размера, например /25. CIDR предполагал изменения как в системе распределения адресного пространства (об этом мы поговорим позже, в разделе «Глобальная система администрирования адресного пространства»), так и в системе маршрутизации.

Последнее было связано с тем, что фактически произошел отказ от концепции классов сетей (A, B, C и D), в которой деление между сетевым адресом и адресом устройства в сети было предопределено. Для маршрутизации CIDR стало необходимым явно указывать, сколько битов IP-адреса относятся к адресу сети (это данные, которые носят название «сетевая маска»).

---

5 RFC 1518: An Architecture for IP Address Allocation with CIDR, URL: https://www.rfc-editor.org/rfc/rfc1518

6 RFC 1519: Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy, URL: https://www.rfc-editor.org/rfc/rfc1519

Вот что отметили в то время члены руководящего комитета IETF — IESG[7]: «CIDR потребует изменения в политике [распределения адресных ресурсов], спецификации протоколов, разработке и внедрении ПО для маршрутизаторов, но не требуется изменение программного обеспечения оконечных устройств». И действительно, новая архитектура была внедрена достаточно быстро. Этому содействовали и относительно небольшой размер Интернета, и его научно-исследовательский характер, и то, что опорная инфраструктура и протоколы находились в стадии разработки.

CIDR позволил избавиться от острых симптомов надвигающейся проблемы, но глобальное решение еще требовалось найти. Поэтому в начале 1994 года IETF начал работу над созданием новой версии протокола IP, позднее получившей название IPv6. Базовая спецификация была опубликована в 1998 году (RFC 2460[8]), а окончательная версия структуры адресации IPv6 — в 2006-м (RFC 4291[9]).

Основные отличия IPv6 от протокола предыдущего поколения — IPv4

Размер адресного пространства

Размер адреса IPv6 составляет 128 бит. Он позволяет адресовать 2^{128} узлов. Это огромное адресное пространство, и масштаб его поистине космический. Например, IPv6 позволяет присвоить 1027 адресов каждой из звезд Млечного Пути. При этом каждая звезда получит адресное пространство в 1018 раз больше, чем весь Интернет IPv4! Очевидно, что дефицита адресов IPv6 в обозримом будущем не предвидится.

IPv6 — это колоссальное количество доступных адресов, это возможность адресации любого мыслимого и немыслимого устройства. Эффективное использование возможностей нового протокола способно породить новый виток информационной революции. Достаточно посмотреть на текущий уровень распределения адресного пространства РИРами (региональными интернет-регистратурами): ясно видно, что их IPv6-пул далек от опустошения (рис. 2).

---

7 RFC 1380: IESG Deliberations on Routing and Addressing, URL: https://www.rfc-editor.org/rfc/rfc1380

8 RFC 2460: Internet Protocol, Version 6 (IPv6), URL: https://www.rfc-editor.org/rfc/rfc2460

9 RFC 4291: IP Version 6 Addressing Architecture, URL: https://www.rfc-editor.org/rfc/rfc4291

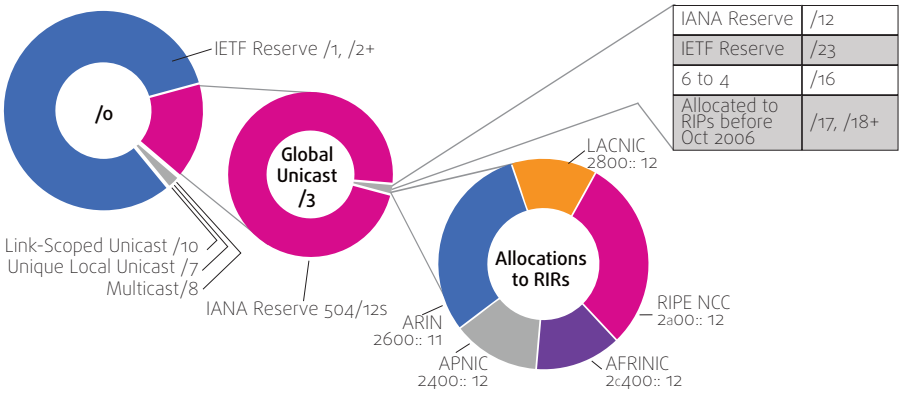

Рис. 2. Адресное пространство IPv6, предоставленное для распределения через региональные интернет-регистратуры. На настоящий момент IPv6-пул РИРов составляет чуть больше пяти блоков /12, и он далек от опустошения; эти блоки составляют ничтожный процент всего доступного в будущем адресного пространства.
Источник: статистика NRO (http://www.nro.net/statistics)

Помимо значительного увеличения адресного пространства изначально предполагалось, что IPv6 сможет поддерживать большее число уровней сетевой иерархии и обеспечит оптимальное распределение адресного пространства с точки зрения маршрутизации и конфигурации. Но в этом отношении ожидания создателей не оправдались: довольно жесткая иерархическая структура адресации была отвергнута операторами в пользу гибкой архитектуры CIDR. Также на сегодняшний момент IPv6 унаследовал многие «болячки» IPv4 (например, независимое от провайдера адресное пространство), которые не способствуют сдерживанию роста таблицы маршрутизации.

Расширяемость и дополнительные функции

При разработке протокола IPv6 особое внимание было уделено возможности добавления новых функций без потери эффективности обработки пакетов на сетевом уровне. IPv6 предполагает наличие дополнительных заголовков для различных расширений (extension header, EH) — например, для криптографической защиты данных (Authentication EH и Encapsulating Security Payload EH). В то же время базовый заголовок IPv6 содержит минимальное число полей и имеет фиксированный размер. В частности, в IPv6 маршрутизаторы не производят фрагментацию, поэтому поля, относящиеся к этой функции, перенесены в соответствующий заголовок расширений (Fragmentation EH).

Как видно из рис. 3, заголовки расширений связаны цепочкой указателей Next Header («Следующий заголовок»).

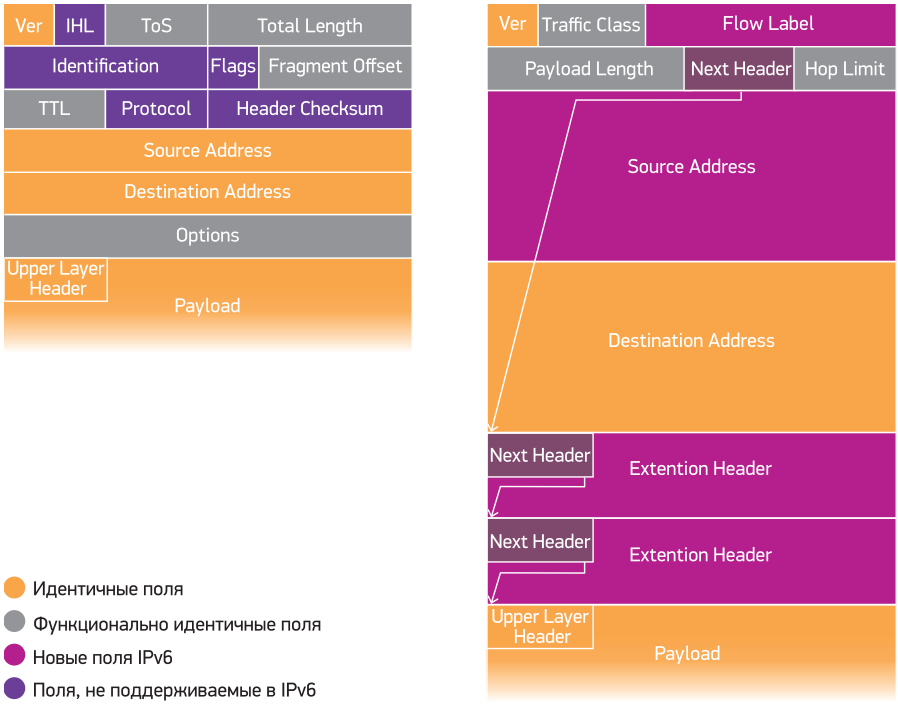

Рис. 3. Форматы пакетов IPv4 и IPv6.

Фрагментация

Как было упомянуто выше, протокол IPv6 иначе обрабатывает фрагментацию пакетов. В случае IPv4, когда маршрутизатор получает пакет, размер которого превышает предел передачи через интерфейс, маршрутизатор производит фрагментацию — дробление пакета на более мелкие части. В дальнейшем они консолидируются получателем в исходный пакет. Заголовок пакета IPv4 имеет соответствующее поле (Fragment Offset), поддерживающее эту функцию.

В IPv6 фрагментация промежуточными устройствами запрещена. Если пакет IPv6 превышает допустимый размер для последующей передачи, маршрутизатор генерирует сообщение ICMP «packet too big» («слишком большой пакет») и посылает его обратно отправителю. В зависимости от приложения отправитель либо выбирает размер пакета, который позволит ему на всем пути следовать без фрагментации, либо дробит пакет самостоятельно. Как и в случае IPv4, консолидация фрагментированных пакетов входит в задачу получателя. Как следствие, передача пакетов IPv6 требует меньших затрат от промежуточного сетевого оборудования.

Автоконфигурация

Для протокола IPv6 была разработана так называемая система автоконфигурации без сохранения состояния (Stateless Autoconfiguration). Данный протокол позволяет различным устройствам, присоединенным к сети IPv6, получить необходимые установки для доступа в Интернет без дополнительных средств — например, без сервиса DHCP (Dynamic Host Configuration Protocol). Суть подхода заключается в том, что устройство получает адрес, состоящий из префикса сети и идентификатора устройства, автоматически сгенерированного с использованием MAC-адреса.

Защита данных

В протокол IPv6 изначально включена система безопасности, основанная на технологии IPsec. Предусмотрено два режима работы: транспортный и туннельный. В транспортном режиме производится защита (шифрование) данных пакета, но не заголовка. С точки зрения маршрутизации такой IP-пакет выглядит вполне обычно, а в задачу получателя входит декодирование содержимого пакета. При использовании туннельного режима данные всего пакета, включая заголовок, шифруются и инкапсулируются в новый пакет. Получатель, указанный в этом новом пакете, является окончанием защищенного канала, или туннеля, и в его задачу входит извлечение изначального пакета и последующая обработка. Дополнительно пакет IPv6 содержит заголовок аутентификации (Authentication EH) для определения подлинности и отсутствия модификации данных пакета.

Мобильность

Поддержка мобильности в протоколах IP означает, что оконечное устройство может изменить свое местоположение в сети и IP-адрес без потери существующих связей, которые соответствуют потокам передачи данных. Для этого мобильные устройства используют отдельные IP-адреса, по которым устройства всегда доступны при передаче данных. За авторизацию мобильного устройства в сети и обеспечение соответствия между реальным и мобильным IP- адресами отвечает «Домашний агент» — устройство, расположенное в «домашней» сети мобильного пользователя. Реализация мобильности в протоколах IPv4 и IPv6 различается. В случае IPv4 передача данных также производится (туннелируется) через «Домашнего агента», в то время как в IPv6 «Домашний агент» обеспечивает только контролирующие функции (авторизацию и обеспечение соответствия между реальным и мобильным адресами). При этом передача данных производится между отправителем и получателем напрямую. Такой подход оптимизирует маршрутизацию данных и, как следствие, повышает качество передачи.

Приведенные особенности протокола IPv6 призваны улучшить производительность, качество и защиту передачи данных. Однако опыт практического внедрения протокола IPv6 показывает, что указанные улучшения весьма незначительны и во многих случаях не используются. Напротив, операторы зачастую прибегают к проверенным методам, разработанным для сетей IPv4. Так, для конфигурации подключенных устройств используется система DHCP, а

в области защиты данных технология IPsec может быть использована в IPv4 почти так же эффективно, как и в IPv6. Эффективная поддержка multihoming (подключения клиента к нескольким сервис-провайдерам для повышения надежности) в IPv6 потребовала отдельного решения и существенно усложнила элегантную структуру маршрутизации, считающуюся одним из преимуществ IPv6. В результате на практике multihoming реализуется аналогично IPv4, что приводит к неоправданному росту таблиц маршрутизации.

Неудивительно, что в среде сетевых операторов существует мнение, что основное преимущество IPv6 — только лишь расширение доступного адресного пространства.

Практика и проблемы внедрения протокола IPv6

Стратегия развития: сосуществование IPv4 и IPv6

Основная проблема перехода от IPv4 к IPv6 — несовместимость двух протоколов. Клиент IPv6 не может напрямую общаться с клиентом, поддерживающим только IPv4.

Изначально представлялось, что эту проблему решит внедрение «двойного стека» — когда компьютеры сети поддерживают оба протокола и подключены как к сети IPv4, так и к сети IPv6. Данное разделение является логическим, а физически используется одна и та же сетевая инфраструктура. Для доступа к ресурсам IPv4 используется протокол IPv4, а к ресурсам IPv6 — протокол IPv6. Все достаточно просто, но... Темпы внедрения IPv6 оказались недостаточными. План «двойного стека» мог сработать, если бы подавляющее большинство компьютеров Интернета имело доступ как к IPv4, так и к IPv6 до того, как пул адресов IPv4 опустошился. В таком случае можно было бы просто отключить поддержку IPv4 и — чудо! — Интернет просто перешел бы на новый протокол. Однако реальность оказалась сложнее.

Сложность внедрения протокола IPv6 во многом связана с так называемым сетевым эффектом. Этот экономический термин описывает явление, когда ценность технологии зависит от числа игроков, ее использующих. Действительно, возможность обмениваться трафиком IPv6 с парой других энтузиастов, как это было в начале 2000-х, с практической точки зрения не представляет особого интереса. Даже при сегодняшнем уровне использования IPv6 (по оценкам Google почти 40% запросов используют этот протокол[10]) большая часть Интернета по-прежнему доступна только через протокол IPv4. Размер этой части Интернета определяет значимость протокола IPv4 и, в обратной пропорции, протокола IPv6 для сервис-провайдеров.

[10] Процент пользователей, использующих IPv6 для доступа к услугам Google, https://www.google.com/intl/en/ipv6/statistics.html#tab=ipv6-adoption

Каждый новый подключенный клиент должен иметь возможность обмениваться данными с Интернетом по протоколу IPv4, что требует предоставления ему адреса IPv4. Скажем прямо, для растущих сервис-провайдеров, возможно, более приоритетным является решение проблемы отсутствия адресов IPv4, чем внедрение IPv6. В то же время важно отметить, что обсуждаемая стратегия и динамика сосуществования двух протоколов основана на предположении, что инфраструктура сервис-провайдера обеспечивает полноценную поддержку IPv6.

Динамика потребности в адресном пространстве IPv4 по мере глобального внедрения IPv6 показана на рис. 4. На нем фиолетовой линией обозначен рост глобального Интернета. По мере внедрения протокола IPv6 доля Интернета, доступного только по IPv4, будет неуклонно уменьшаться (оранжевая кривая). Синяя линия отображает размер сервис-провайдера, характеризуемый, например, числом подключенных пользователей. В данном случае рассматривается растущий провайдер. Наконец, потребность в адресах IPv4 показана кривой зеленого цвета.

По мере расширения клиентской базы провайдера пропорционально увеличивается потребность в дополнительных адресах IPv4. В то же время все большая и большая часть Интернета становится доступной по протоколу IPv6, что выражается в обратной тенденции, когда все меньшее число пользовательских соединений основано на протоколе IPv4. Соответственно, потребность в адресах IPv4 снижается. Наконец, когда подавляющее большинство ресурсов Интернета станет доступным по IPv6, потребность в IPv4 станет ничтожной. Таким образом, завершится фаза перехода Интернета на протокол IPv6. Продолжительность этой фазы может составить несколько лет. Не исключена, правда, вероятность, что данная фаза не закончится никогда, но об этом — чуть позже.

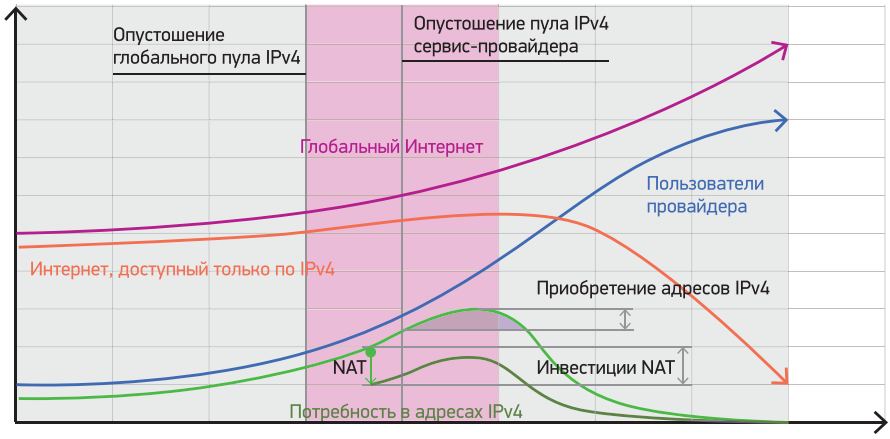

Рис. 4. Динамика сосуществования IPv4 и IPv6.

Как видно из графика, наиболее критичной фазой для сервис-провайдера является промежуток времени с момента опустошения глобального свободного пула IPv4 до момента, когда потребность в дополнительных адресах IPv4 начнет уменьшаться. Эта фаза отмечена на графике сиреневым цветом.

Надо заметить, что высота порога, образуемого зеленой кривой, для разных провайдеров отличается. Также различен момент завершения свободных адресов в собственном пуле провайдера (вторая вертикальная синяя линия). Другими словами, умеренно растущий провайдер с достаточным запасом свободных адресов имеет шансы «перезимовать» переходный период без особых ухищрений. Важно отметить, что и в этом случае необходимой является полноценная поддержка IPv6 в инфраструктуре провайдера и неуклонное массовое распространение IPv6 в глобальном Интернете. Все большее число сервис-провайдеров страдают от проблемы нехватки IPv4.

Существует два способа решения этой проблемы. Первый — это получение дополнительных адресов IPv4. Однако в соответствии с текущей политикой распределения оставшегося адресного пространства IPv4[11] максимум, на что может рассчитывать провайдер, — это одноразовый блок размером /24, да и то придется подождать, пока такой блок появится, например, вследствие возврата адресов закрывшегося оператора. Рассчитывать на это не стоит – на февраль 2023 года список ожидания превысил 1200 претендентов, и уже больше года свободных блоков не появлялось.

Поэтому более практичный вариант получения дополнительных адресов – это их покупка на рынке IPv4. Уже несколько лет на рынке работают так называемые брокеры, связывающие желающих продать и купить. Передача адресов от одной организации к другой регламентирована соответствующими политиками RIPE – «Передача цифровых интернет-ресурсов и изменение официального юридического имени члена»[12] и «Передача цифровых интернет-ресурсов между региональными интернет регистратурами»[13]. Однако следует иметь в виду, что покупка адресов – дорогостоящая операция. За последние два года цена в расчете на один адрес удвоилась и достигла $45-55. Хотя наблюдается некоторая стабилизация и даже охлаждение рынка в 2022, скорее всего, это явление временное, и цены продолжат расти, см. рис. 5.

[11] IPv4 Address Allocation and Assignment Policies for the RIPE NCC Service Region, секция 5.1, https://www.ripe.net/publications/docs/ripe-733

[12] Transfer of Internet Number Resources and Change of a Member's Official Legal Name, RIPE-758, 30 марта 2021 г., https://www.ripe.net/publications/docs/ripe-758

[13] Inter-RIR Transfer of Internet Number Resources, RIPE-769, 24 ноября 2021 г., https://www.ripe.net/publications/docs/ripe-769

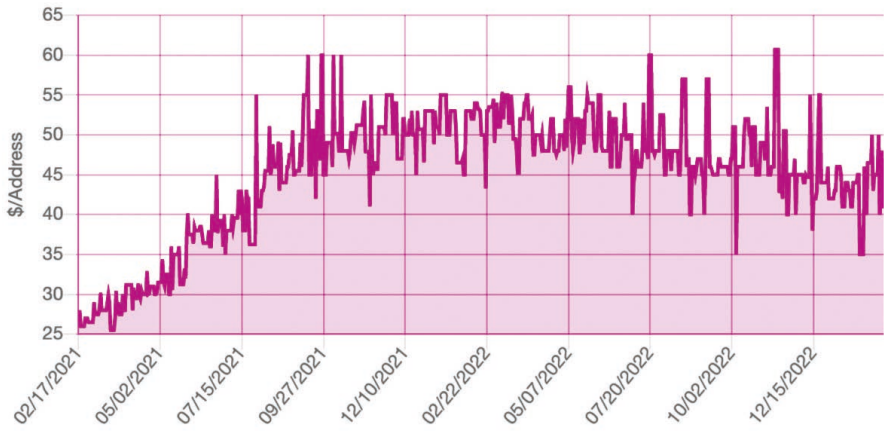

Рис 5. Колебание цен на адреса IPv4 в пересчете на один адрес.
Источник: https://ipv4.global/

В этой связи второй способ — повышение эффективности использования адресного пространства с помощью технологии NAT (Network Address Translation) — является более реальной альтернативой или дополнительным решением. Этот сценарий показан на графике кривой розового цвета.

Поскольку мы заговорили о технологии NAT, пожалуй, стоит остановиться на ней поподробнее. Ведь эта технология является ключевой в моделях сосуществования IPv4 и IPv6.

Техническое отступление: как происходит передача данных в Интернете

Прежде чем перейти непосредственно к разговору о будущем Интернета и перспективах IPv6, давайте совершим краткий экскурс в техническую область и в общих чертах рассмотрим, как же происходит передача данных в Интернете и какую роль играют адреса. Работа Интернета основана на технологии пакетной коммутации без установления соединения. Структура пакета определена протоколом IP, при этом каждый пакет содержит IP-адрес отправителя и получателя. В задачу каждого узла сети (также называемого маршрутизатором) входит передача пакета, полученного от соседнего узла, к последующему узлу. Выбор каждого следующего узла происходит с помощью системы маршрутизации. Благодаря этой системе маршрутизатор знает, какому из своих соседей следует передать пакет с конкретным IP-адресом получателя.

Однако с точки зрения пользователя передача данных происходит между его приложением и приложением получателя. Например, между веб-браузером и веб-сайтом. Поэтому можно представить, что существует виртуальное соединение между этими приложениями: по нему и происходит передача данных. Помимо IP-адреса отправителя (в данном случае — компьютера пользователя) и IP-

адреса получателя (веб-сервера) это соединение характеризуется дополнительными параметрами — так называемыми портами получателя и отправителя. Их можно рассматривать как локальные идентификаторы конкретных приложений на компьютере. Наконец, транспортный протокол (например, TCP или UDP) является пятым параметром, однозначно определяющим поток данных в Интернете в пределах ограниченного времени.

Таким образом, отправитель и получатель данных в действительности каждый адресуются парой {IP-адрес, порт}. Именно эта особенность используется в технологии NAT (Network Address Translation), или более точно — NAPT (Network Address & Port Translation). С помощью одного IP-адреса можно теоретически адресовать 65 535 «соединений» — число, значительно превышающее потребности единичного пользователя. В этом случае устройство NAT для внешней сети будет выглядеть как компьютер с очень большим числом одновременно работающих приложений. Хотя на самом деле устройство NAT при передаче пакетов подставляет вместо порта и собственного IP-адреса (как адреса получателя с точки зрения внешних приложений) порт и локальный IP-адрес реального получателя. Обычно для адресации конечных устройств локальной сети, расположенной за устройством NAT, используется специальное зарезервированное адресное пространство. Схема работы NAT показана на рис. 6.

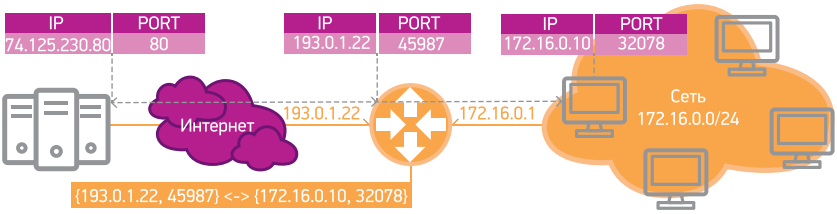

Рис. 6. Схема работы NAT.

Насколько эффективен NAT? Это зависит от характера приложений, работающих на конечных устройствах, и интенсивности их взаимодействия с глобальным Интернетом. На сегодня компьютер обычного пользователя во время работы в сети создает от 60 до 100 соединений с различными ресурсами глобального Интернета. Цифра может показаться большой, но ведь многие приложения открывают более одного соединения — так функционирует большинство веб-приложений. Например, Google Maps одновременно использует несколько десятков соединений. Но даже если эта цифра на порядок крупнее, трудностей не возникает: технология NAT позволяет совместно использовать один и тот же IP-адрес более чем 60 пользователям.

Звучит очень привлекательно — но, к сожалению, в реальности все не так радужно. Технология NAT содержит ряд серьезных недостатков, о которых мы поговорим позже. Здесь же отметим, что NAT нарушает принцип «прозрачности»

соединений между любыми конечными устройствами в Интернете. Помимо усложнения архитектуры сети, для полноценной работы некоторых приложений требуются дополнительные средства, такие как STUN[14], ICE[15] , TURN[16]. Использование каскадов NAT, когда в сети за устройством NAT расположены еще и NAT со «вложенными» сетями, только усугубляет эти проблемы.

Переходные технологии сосуществования

Итак, технология NAT-мультиплексирования — еще один метод решения проблемы сосуществования двух протоколов, позволяющий бороться с острой нехваткой адресов IPv4. Однако по-прежнему одним из основных препятствий перехода к IPv6 является его несовместимость со своим предшественником — протоколом IPv4. Устройство, поддерживающее только IPv6, не может непосредственно обмениваться данными с устройством IPv4. Виной этому является, скорее, протокол IPv4, который был разработан для адресации нескольких десятков, может быть — сотен или тысяч устройств Сети и не предусматривал способа расширения.

Переходный план «двойного стека» предполагал отсутствие устройств, «говорящих» только на одном из протоколов, другими словами — глобальное двуязычие. Соответственно, он основан на предположении, что все рассматриваемые устройства имеют адреса IPv4. Опустошение пула адресов IPv4 существенно ограничило сферу применения этого подхода. Поэтому для обмена данными между устройствами и сетями разных протоколов необходимо применение дополнительных технологий — так же как мы прибегаем к услугам переводчика для преодоления языкового барьера.

За последние два десятилетия было предложено множество решений, но не все оказались эффективными и устойчивыми, а многие вообще не прижились.

Давайте посмотрим, что же имеется в арсенале сервис-провайдеров на сегодняшний день. Помимо «двойного стека», предполагающего прозрачную связность, переходные технологии делятся на два типа: туннеллирование и трансляцию.

Технологии туннелирования

Технологии туннелирования приходят на помощь, когда инфраструктура сервис-провайдера не поддерживает один из протоколов.

---

[14] RFC 5389: Session Traversal Utilities for NAT (STUN),
URL: https://www.rfc-editor.org/rfc/rfc5389

[15] RFC 5245: Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,
URL: https://www.rfc-editor.org/rfc/rfc5245

[16] Traversal Using Relay NAT,
URL: https://ru.wikipedia.org/wiki/Traversal_Using_Relay_NAT;
RFC 5766: Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN),
URL: https://www.rfc-editor.org/rfc/rfc5766

6rd

«rd» в названии этой технологии означает «rapid deployment», или быстрое развертывание. И действительно, эта технология позволила одному из крупнейших французских провайдеров Free в течение пяти недель осуществить поддержку IPv6 для пользователей сети. Технология была после этого стандартизована в IETF[17]. 6rd делает доступным Интернет IPv6 пользователям провайдера широкополосного доступа, не требуя при этом поддержки IPv6 в сети самого провайдера.

Во-первых, сеть 6rd использует собственное адресное пространство IPv6, полученное от региональной интернет-регистратуры. Это позволяет сервис-провайдеру анонсировать реальные IPv6-префиксы и, таким образом, более точно определять собственную политику маршрутизации.

Во-вторых, вся зона функционирования 6rd ограничена сетью сервис-провайдера. Так называемые шлюзы 6rd встроены в оконечное оборудование клиента (CPE, customer premise equipment), а релеи являются частью инфраструктуры сервис-провайдера.

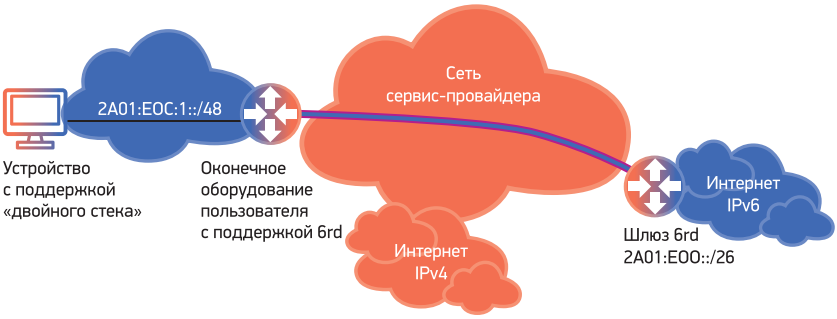

Рис. 7. Схема работы технологии 6rd.

DS-Lite

DS-Lite[18] в некотором смысле является зеркальной технологией по отношению к 6rd. DS-Lite предполагает, что сеть провайдера полностью поддерживает IPv6, а туннели используются для передачи трафика IPv4 от сети пользователя к устройствам NAT сервис-провайдера. Также подразумевается, что устройства сети пользователя поддерживают «двойной стек», а именно оба протокола IPv4 и IPv6.

Суть метода заключается в одновременном применении технологий туннелирования (инкапсуляция трафика IPv4 в пакеты IPv6) и централизованного NAT,

[17] RFC 5969: IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) - Protocol Specification, URL: https://www.rfc-editor.org/rfc/rfc5969

[18] RFC 6333: Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion, URL: https://www.rfc-editor.org/rfc/rfc6333

или CGN (Carrier Grade NAT, также называемого LSN, Large Scale NAT). Благодаря этому ограниченный пул адресов IPv4 совместно используется всеми пользователями сервис-провайдера. Обмен трафиком с интернет-ресурсами IPv4 происходит с использованием протокола IPv4, а с ресурсами IPv6 — с использованием IPv6. Эта схема не предусматривает трансляции протоколов.

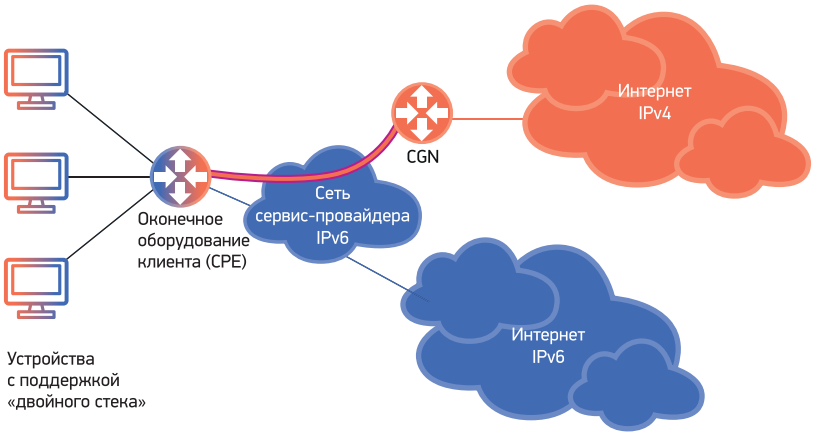

Рис. 8. Схема работы DS-Lite.

На рис. 8 представлена схема работы DS-Lite. Как можно заметить, обмен трафиком с ресурсами IPv6 происходит непосредственно, без использования каких-либо промежуточных технологий, например, туннелей.

В отношении IPv4 ситуация гораздо сложнее. Нехватка адресного пространства IPv4 — это серьезная проблема для растущего числа сетей. Поэтому схемы, предусматривающие назначение каждому абоненту публичного адреса IPv4, используемого устройством NAT-пользователя для построения домашней локальной сети, имеют все более ограниченное применение.

Возможным решением этой проблемы (кстати, уже применяемым некоторыми сервис-провайдерами) является создание еще одного уровня NAT в сети сервис-провайдера. Такая схема работает в общем случае, но результат ее применения — существенные ограничения для многих сегодняшних и будущих приложений, а также сложность обслуживания.

Задача DS-Lite — исключить каскадирование устройств NAT, когда все устройства пользователей непосредственно взаимодействуют с центральным устройством NAT сервис-провайдера. В этом случае оконечное устройство пользователя не выполняет никаких функций NAT, а вместо этого обеспечивает создание туннелей к центральному NAT для каждого нового соединения между приложениями пользователя и сервисами Интернета.

Таким образом, все пользовательские соединения, так же, как и в схеме каскадирования NAT, отображаются центральным CGN. Однако значительно по-

вышается прозрачность архитектуры, растет эффективность использования адресного пространства IPv4.

Кстати, о прозрачности. Одна из основных проблем, связанных с применением NAT, — это контроль приложений за значениями порта и IP-адреса соединений, поскольку устройство NAT заменяет их на динамически присваиваемые. От этого зависит нормальное функционирование некоторых приложений, например, большинства мультимедийных интерактивных программ. На сегодняшний день разработано несколько механизмов решения этой проблемы — такие как STUN, ICE и TURN. Но очевидно, что каскадирование устройств NAT усложняет ситуацию.

В то же время, поскольку централизованная трансляция адресов осуществляется для каждого сетевого потока, DS-Lite сложно масштабировать, особенно в сетях крупных провайдеров. Чтобы решить эту проблему, в IETF было предложено расширение DS-Lite, получившее название «Lightweight 4over6», или lw4o6. Новый подход требует поддержки состояния не для каждого потока, а только для каждого абонента и перемещает функцию NAT обратно на клиентское оборудование (CPE). Технология lw4o6 была также стандартизована в IETF[19].

Отметим, что технологии DS-Lite и lw4o6 не предусматривают поддержку устройств, работающих только по протоколу IPv6. Для этого используются технологии трансляции.

Технология трансляции: NAT64 + DNS64

Логично предположить, что в недалеком будущем появятся устройства, поддерживающие только IPv6. Если мы говорим о масштабных мобильных, сенсорных или RFID-сетях, необходимость поддержки двух протоколов усложнит и удорожит такие устройства.

Для взаимодействия таких сетей с Интернетом IPv4 необходимо применение трансляции адресов IPv6 в адреса IPv4 и обратно. Ввиду недостатка ресурсов IPv4 здесь, как и в случаях, рассмотренных выше, необходимо применение мультиплексирования потоков. По существу, нужно использовать технологию централизованного NAT с внедрением дополнительной функции трансляции протоколов. Этот компонент еще называют NAT64[20]. Взаимодействие с другими сетями IPv6 происходит прозрачно: эта архитектура показана на рис. 9.

Однако в данной схеме есть одна особенность, а именно необходимость дополнительной поддержки одного из наиболее критических приложений Интернета — системы доменных имен DNS.

---

[19] RFC7596: Lightweight 4over6: An Extension to the Dual-Stack Lite Architecture, URL: https://www.rfc-editor.org/rfc/rfc6146

[20] RFC 6146: Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers, URL: https://datatracker.ietf.org/doc/rfc6146

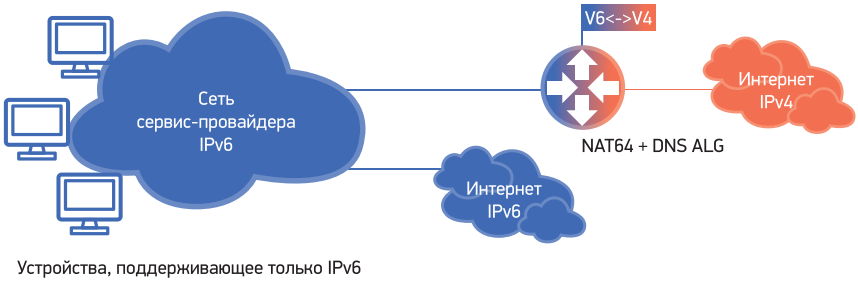

Рис. 9. Архитектура системы трансляции протоколов NAT64.

Дело в том, что для большей части ресурсов Интернета запрос DNS вернет адрес IPv4. Поскольку сети, о которых идет речь, поддерживают только IPv6, такой ответ DNS вряд ли окажется полезным. Для решения этой проблемы используется дополнительный компонент — шлюз приложений (Application Layer Gateway, ALG). Суть его заключается в замещении адреса IPv4 в ответе DNS на синтезированный адрес IPv6, который понятен и клиенту, и транслятору протоколов NAT64.

Работа DNS ALG происходит следующим образом. Как обычно, перед началом связи клиент посылает запрос локальному DNS-серверу. В нашем случае его роль выполняет ALG. Он производит разрешение запроса и, допустим, получает IPv4-адрес искомого ресурса. Но в ответ клиенту ALG подставляет синтезированный адрес IPv6. По существу, этот адрес состоит из предустановленного префикса (известного и ALG, и NAT64), а также из IPv4-адреса ресурса. Теперь, когда клиент попытается установить связь с ресурсом, NAT64 поймет, что клиент использует синтезированный адрес, и преобразует его в исходный IPv4-адрес получателя. Схематически это показано на рис. 10.

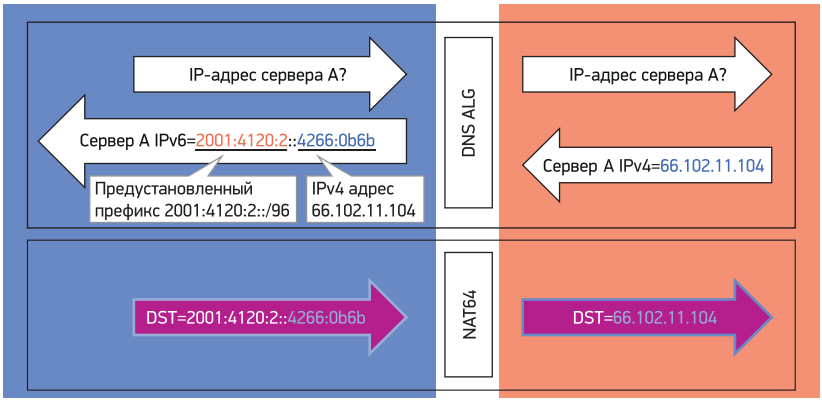

Рис. 10. Схема работы DNS ALG.

Вопросы внедрения IPv6 в мобильных сетях

Сегодня разговор об эволюции Интернета немыслим без взгляда на мобильные сети. Здесь мы наблюдаем наиболее стремительный рост как по количеству абонентов, так и по возможностям, которые они открывают для пользователей. Архитектурные решения, принимаемые при разработке или модернизации мобильных сетей, определяют архитектуру будущего Интернета. Поставим вопрос более остро: останется ли Интернет уникальной коммуникационной средой с колоссальным инновационным потенциалом или наше информационное пространство будут определять закрытые и ограниченные платформы мобильных приложений, такие как Apple Store или Google Play?

Развитие мобильных сетей началось с сетей мобильной телефонии, основанных на телефонных стандартах и технологии коммутации каналов. Передача данных была внедрена позже как отдельная подсистема, существенно отличающаяся как по архитектуре, так и по используемым технологиям. Так, для обеспечения услуг на основе пакетной передачи — в первую очередь для доступа к Интернету — в сетях 2G и 3G была разработана система GPRS (General Packet Radio Service). Для возможности предоставления услуг голосовой связи на основе протокола IP в 2002 году была разработана система IMS (IP Multimedia System). Сегодняшние 3G-сети предоставляют услуги передачи данных и доступа в Интернет в качестве стандартного пакета, однако для осуществления голосовой связи, как правило, по-прежнему используются сети коммутации каналов.

Появление сетей следующего поколения LTE/4G существенно изменило ситуацию. Действительно, эти сети используют исключительно технологию пакетной передачи на основе протокола IP. Для оператора это означает возможность унификации передачи голоса и данных. И хотя для связи с традиционными телефонными сетями необходимы шлюзы, связь между абонентами собственной сети и сетями партнеров, также использующих эти технологии, а также предоставление доступа в Интернет осуществляется унифицированной инфраструктурой на основе пакетной коммутации IP.

В архитектуре LTE опорная сеть, так называемая EPC (Evolved Packet Core), представляет собой нормальную сеть пакетной коммутации на основе IP. Дополнительные устройства и шлюзы необходимы для контроля доступа к услугам, поддержки мобильности и роуминга, а также биллинга. Схематично структура сети LTE приведена на рис. 11.

Так, узел управления мобильностью MME (Mobility Management Entity) осуществляет контроль доступа к сети LTE, производит аутентификацию пользователя и поддерживает функции мобильности. Обслуживающий шлюз SGW (Serving Gateway) является маршрутизатором доступа, он поддерживает в том числе и переход мобильного терминала между базовыми станциями (eNodeB). Наконец, пакетный шлюз (PGW, Packet Data Network Gateway) является граничным маршрутизатором, обеспечивающим связность с другими системами: IMS и внешними сетями, например Интернетом.

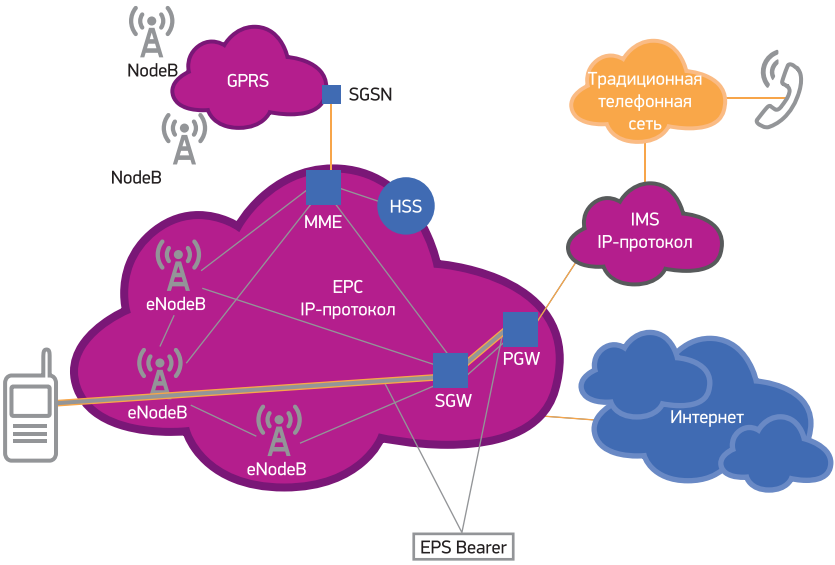

Рис. 11. Архитектура сети LTE.

Здесь стоит отметить одну архитектурную особенность мобильной сети. Передача данных во внутренней инфраструктуре происходит по виртуальному каналу, или туннелю, соединяющему мобильное пользовательское устройство, например телефон, с пакетным шлюзом PGW. В сетях LTE такие туннели называются носителями EPC, а в сетях GPRS использовался термин «PDP-контекст». Специальные протоколы отвечают за создание виртуального канала и резервирования определенных ресурсов сети, в том числе самого ценного ресурса — радиоканала.

Поскольку стандарт LTE поддерживает только пакетную коммутацию IP, потребовалась новая структура сети передачи голоса. Ведь, как уже упоминалось, в мобильных сетях предыдущих поколений голосовая связь была основана на коммутации каналов, а передача данных являлась сопутствующей подсистемой.

Одним из наиболее перспективных подходов обеспечения голосовой связи является архитектура VoLTE (Voice over LTE — «Голос поверх LTE»). Этот подход использует подсистему IMS (IP Multimedia Subsystem), полностью основанную на IP и использующую стандарты передачи голоса в IP-сетях — SIP и RTP. В результате голосовые и управляющие соединения представляют собой не что иное, как обычные потоки данных сети LTE.

Итак, на повестку дня достаточно остро встает задача внедрения IMS. Но есть и еще один важный вопрос: а какой из протоколов применять — IPv4 или IPv6? С одной стороны, так же, как и в сетях фиксированной связи, протокол IPv4 хорошо отработан и поддерживается всеми производителями оборудования.

С другой стороны, нехватка адресного пространства IPv4 заставляет операторов применять более сложные системы трансляции и динамического назначения адресов. К тому же следует учитывать колоссальные масштабы мобильных сетей, насчитывающих миллионы абонентов. Ведь даже использование зарезервированных адресных блоков, например, 10.0.0.0/8, наталкивается на серьезные ограничения. Действительно, в рамках этого блока, широко используемого в сетях с трансляцией адресов, максимальное число адресуемых устройств составляет всего 16,7 миллиона — для многих операторов это меньше, чем число абонентов. Все перечисленные факторы заставляют мобильных операторов серьезно задуматься об использовании протокола IPv6.

Но, как мы уже обсуждали, внедрение IPv6 само по себе не является панацеей — на сегодняшний день Интернет в основном доступен по протоколу IPv4. То есть, хотя внедрение IPv6 является разумной долгосрочной стратегией, на текущем этапе необходимо также обеспечить доступ к сетям и ресурсам IPv4, а значит — использовать технологии сосуществования.

Внедрение IPv6 в мобильных сетях на основе технологии трансляции XLAT

В предыдущих разделах мы рассмотрели различные технологии сосуществования — изначальную стандартную архитектуру «двойного стека», технологии туннелирования и трансляции. До реализации возможности создания виртуальных каналов (PDP-контекст в GPRS и носитель EPC в LTE), поддерживающих «двойной стек», данная архитектура требовала дублирования туннелей и, соответственно, сетевых ресурсов. Поэтому до появления релизов R8 и R9, в которых такие соединения были определены для EPC и GPRS соответственно, этот подход считался неприемлемым. Сегодня — это рабочая альтернатива, серьезно рассматриваемая некоторыми операторами.

Однако поддержка одновременно обоих протоколов в базовой системе, включая EPC (в сетях 3G — GPRS) и IMS, означает существенное удорожание инфраструктуры и обслуживания. Также серьезной проблемой остается нехватка адресов в зарезервированных блоках, требующая разбиения сети на отдельные домены и контроля отсутствия адресных конфликтов. Более подробное обсуждение вопросов внедрения архитектуры «двойного стека» в мобильных сетях и связанных с этим проблем можно найти в RFC 6459[21].

В связи с этим более реалистичным представляется подход, основанный на архитектуре трансляции, при котором опорная сеть и базовые системы поддерживают только один протокол — IPv6. Мобильные устройства при этом поддерживают оба протокола, а поддержка приложений IPv4 происходит с помощью трансляции IPv4 в IPv6, передачи данных опорной сетью и обратной трансляции шлюзом взаимодействия с сетями IPv4.

[21] RFC 6459: IPv6 in 3rd Generation Partnership Project (3GPP) Evolved Packet System (EPS), URL: https://www.rfc-editor.org/rfc/rfc6459

Эта архитектура, получившая название 464XLAT и описанная в RFC 6877[22], во многом похожа на уже рассмотренную нами архитектуру DS-Lite. Основным отличием является то, что при передаче данных по IPv6-сети провайдера происходит адресная трансляция, а не туннелирование. Данный подход отличается относительной простотой, а кроме этого, более эффективно используется полоса пропускания — важный фактор в беспроводных сетях. Схема этой архитектуры приведена на рис. 12.

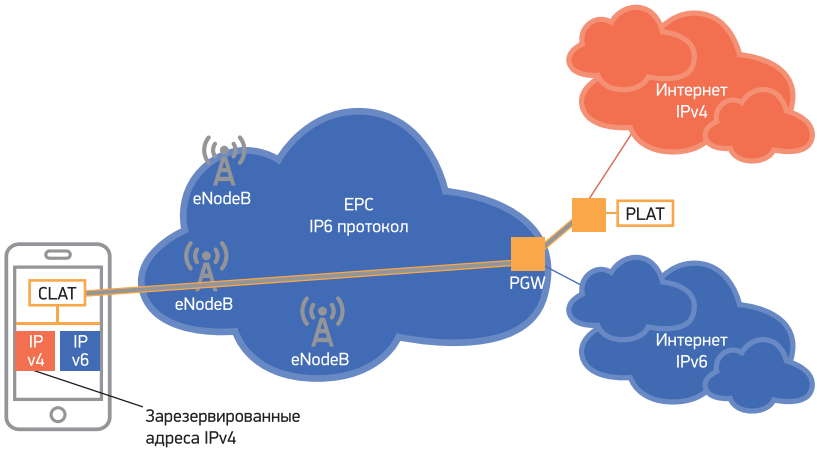

Рис. 12. Архитектура 646XLAT с опорной сетью, поддерживающей только IPv6.

В то время как передача данных IPv6 происходит в сетях 464XLAT абсолютно прозрачно, для передачи трафика IPv4 производится двойная трансляция. Сначала производится трансляция IPv4 в IPv6 на стороне клиента. Для этого используется так называемое устройство CLAT (customer-side translator, транслятор со стороны клиента). В мобильной сети функция CLAT реализована в пользовательском оборудовании — мобильном телефоне. CLAT является транслятором без сохранения состояния, что существенно упрощает его реализацию и работу. Дело в том, что все адресное пространство IPv4 может быть однозначно отображено в часть адресного пространства IPv6, а точнее, в пространство, определяемое IPv6-префиксом длиной 96 бит (/96).

На другой стороне IPv6-сети провайдера производится обратная трансляция. При этом используется тот же алгоритм преобразования, только в обратную сторону. Другими словами, если IPv6-адрес получателя 2001: DB8: AAAA::198.51.100.1, то соответствующий ему адрес IPv4 — 198.51.100.1. Этой задачей занимается устройство PLAT (provider-side transpator, транслятор со стороны провайдера).

[22] RFC 6877: 464XLAT: Combination of Stateful and Stateless Translation, URL: https://www.rfc-editor.org/rfc/rfc6877

Во многих случаях PLAT выполняет также функцию стандартного транслятора NAT. Дело в том, что для IPv4-адресации мобильных терминалов многие операторы используют зарезервированные адресные блоки (например, 10.0.0.0/8). При передаче пакета в Интернет такие адреса транслируются в глобальные адреса из пула провайдера. Напомним, что NAT является устройством, сохраняющим состояние, и поэтому более сложным и дорогостоящим, чем CLAT.

В архитектуре 464XLAT ограничение размера пула зарезервированных адресов не является проблемой, поскольку каждый CLAT может быть однозначно идентифицирован уникальным IPv6-префиксом, назначенным ему оператором, например, при подключении к сети.

Хотя существуют убедительные примеры использования этой архитектуры в сетях мобильных операторов, многих останавливает недостаточная ее поддержка ведущими разработчиками операционных систем для смартфонов. В настоящее время только Android продолжает поддерживать трансляцию XLAT.

Вопросы роуминга при внедрении IPv6

У различных мобильных операторов уровень поддержки и стратегия внедрения IPv6 могут существенно отличаться. Поэтому обеспечение роуминга требует тщательного анализа вероятных проблемных ситуаций.

Роумингом называется возможность предоставления услуг сотовой связи абоненту вне зоны обслуживания его «домашней» сети другим оператором, так называемой гостевой сетью. При этом абоненту не требуется заключать договор с принимающим оператором, а плата за услуги взимается «домашним» оператором. Услуга роуминга требует предварительной взаимной договоренности между операторами.

В общих чертах роуминг осуществляется следующим образом. При включении мобильного устройства вне домашней сети оно просканирует все радиоканалы в поиске сети, к которой можно подключиться. При подключении узел MME (или узел SGSN в случае 3G/GPRS сети) сначала сделает запрос в домашнюю сеть пользователя к серверу HSS (или HLR в случае 3G) для получения профиля абонента и последующей его аутентификации. Профиль абонента, помимо прочего, содержит информацию о варианте маршрутизации и доступных типах PDP-контекстов (носителей EPC в сетях LTE). По завершении процесса регистрации возможно создание PDP-контекста. Здесь, в зависимости от конфигурации абонента, возможны два варианта: маршрутизация через домашнюю сеть и маршрутизация с местным выходом.

В первом случае при активации контекста PDP устройству абонента будет назначен IP-адрес из домашней сети. Маршрутизация всего трафика будет проходить через домашнюю сеть. Во втором варианте IP-адрес назначается из гостевой сети, соответственно, и трафик во внешние сети, например

Интернет, будет передаваться из гостевой сети, без захода в домашнюю сеть. Тем самым может быть достигнут оптимальный маршрут. Различие между этими двумя вариантами показано на рис. 13.

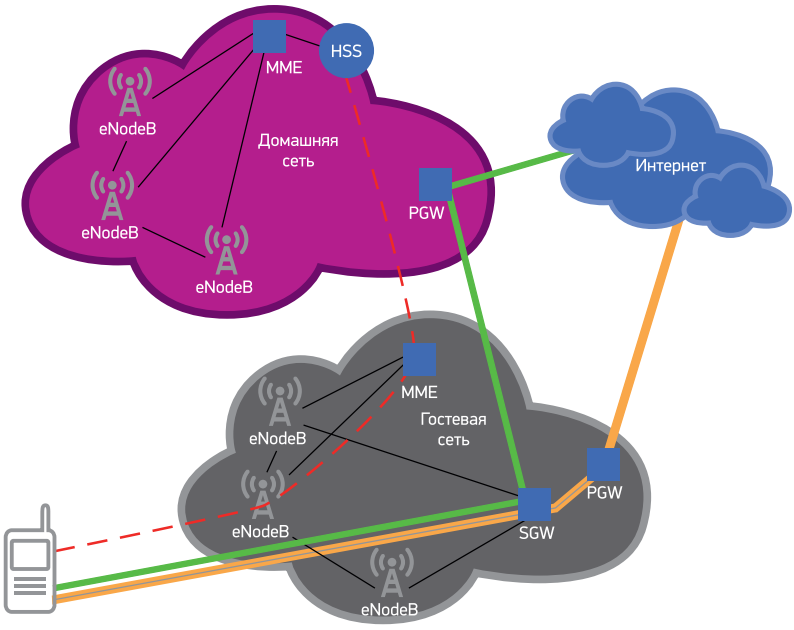

Рис. 13. Два варианта маршрутизации трафика при роуминге: зеленым цветом показан PDP-контекст при маршрутизации через домашнюю сеть, оранжевым — при маршрутизации с местным выходом; красная пунктирная линия отображает канал сигнализации.

Стандарты 3GPP определяют три типа PDP-контекста: PDP/PDN типа IPv4, PDP/PDN типа IPv6 и PDP/PDN типа IPv4v6. Последний тип контекста был введен в спецификации начиная с Релиза 9 для поддержки архитектуры «двойного стека».

Проблема при подключении абонента к гостевой сети может возникнуть, если данная сеть не поддерживает контекста IPv4v6 и, соответственно, не может правильно интерпретировать соответствующую запись в профиле абонента. В этом случае абоненту будет отказано в подключении. Одним из решений этой проблемы является определение нескольких доступных типов PDP-контекста для абонента, например, IPv4v6 и IPv4. В ответ на запрос на роуминг из гостевой сети для обеспечения максимальной совместимости оператор передаст профиль абонента только с типом PDP-контекста IPv4. В то же время в домашней сети абонент может пользоваться более эффективным PDP-контекстом — IPv4v6. Оператор может использовать и так называемые белые листы — списки гостевых сетей, для которых поддержка определенных PDP-контекстов заведомо известна.

Другой класс проблем связан с назначением IP-адреса мобильному устройству при маршрутизации с местным выходом. В этом случае возможно несоответствие между запрашиваемым и доступным PDP-контекстами, между возможностями приложений и подсистем, например IMS, и созданными типами каналов, а также между функциями устройства и соответствующими функциями сети, например при использовании модели 464XLAT, описанной в предыдущем разделе. Решениями данных проблем является запрещение варианта маршрутизации с местным выходом или адаптация профиля абонента в соответствии с возможностями гостевой сети. В последнем случае могут также применяться белые списки.

Вопросы безопасности, связанные с IPv6

IPv6 является относительно новым протоколом, во многом отличным от своего предшественника — IPv4. Соответственно, с внедрением IPv6 связаны дополнительные риски.

Приведем такой пример. Изначально спецификация IPv6 определяла заголовок расширений маршрутизации (Routing EH), а также его подтип Routing Header Type 0, или RH0. Поле RH0 может содержать множество адресов промежуточных узлов, через которые должна пройти передача пакета, причем один и тот же адрес может быть указан более одного раза. А значит, есть вероятность того, что пакеты будут осциллировать между двумя узлами, тем самым вызывая перегрузку канала между ними. Эта возможность может быть использована атакующим для создания атаки отказа в обслуживании (Denial of Service, DoS) с усилением. Поэтому в 2007 году IETF исключил данную функциональность из спецификации IPv6.[23]

В целом, риски можно разделить на следующие категории:

Риски, связанные с недостаточной подготовленностью персонала

Очевидно, что недостаточный опыт и подготовка персонала в обнаружении и решении проблем, связанных с IPv6, а также недостаточно хорошее понимание различных новых функций являются существенными рисками безопасности.

Риски, связанные с неадекватной политикой безопасности в отношении IPv6

Многие организации по-прежнему рассматривают протокол IPv6 как экспериментальный, даже если его внедрение происходит в рабочей инфраструктуре. Как следствие, политика безопасности зачастую разрабатывается и исполняется менее строго. Это усугубляется тем, что во многих случаях механическая репликация существующей политики для IPv4 невозможна — это связано как с различиями в семантике IPv6, так и с возможностями оборудования.

---

[23] RFC 5095: Deprecation of Type 0 Routing Headers in IPv6, URL: https://www.rfc-editor.org/rfc/rfc5095

Один из примеров — неэффективность использования экранов безопасности без сохранения состояния (stateless firewals). Дело в том, что заголовок IPv6 не содержит поля, указывающего на протокол верхнего уровня, например, TCP. В IPv6 протокол верхнего уровня определяется последним заголовком расширений — заголовком верхнего уровня (Upper Layer Header, ULH). Экраны безопасности обычно содержат правила, основанные как на информации интернет-уровня (IP), так и на протоколах верхнего уровня (например, TCP). Поскольку для фрагментов информация о последнем будет отсутствовать, возникает неопределенность в обработке таких пакетов экраном. Но даже если пакет доставлен целиком, для получения информации о протоколе верхнего уровня устройству потребуется проанализировать все заголовки расширений IPv6, которые в пакете представлены в виде связанного списка. Очевидно, что в некоторый случаях это может привести к значительным дополнительным затратам на обработку, уменьшая производительность устройства.

Риски, связанные с новой функциональностью IPv6 и новыми векторами атаки

Выше мы упомянули обнаруженную уязвимость протокола при применении специальных расширений, связанных с маршрутизацией пакета. Но атакующий может использовать и малтикаст-адреса пакетов для сканирования локальной сети на предмет присутствующих там устройств. Были обнаружены новые векторы атак, связанные и с системой автоконфигурации. Подробное обсуждение этих проблем можно найти в следующих документах IETF[24].

Уже упомянутые заголовки расширений ввиду их многообразия и сложности разбора также несут существенные риски, начиная от меньшей эффективности устройств безопасности и заканчивая возможностью их использования в качестве атакующего средства. Более подробно эта проблематика рассмотрена в документе IETF[25].

Текущее состояние внедрения и использования IPv6

С момента публикации спецификации IPv6 прошло уже больше 25 лет, но пока рано говорить о полном замещении протокола IPv4 протоколом IPv6. В то же время тенденция в целом положительная. Например, процент пользователей, использующих IPv6 для доступа к услугам Google, в феврале 2024 года составил 40-45%. Стоит заметить, что 10 лет назад эта цифра составляла меньше 3% (см. рис. 14).

---

[24] RFC 3756: IPv6 Neighbor Discovery (ND) Trust Models and Threats, URL: https://www.rfc-editor.org/rfc/rfc3756;
RFC 6104: Rogue IPv6 Router Advertisements, URL: https://www.rfc-editor.org/rfc/rfc6104

[25] RFC 9098: Operational Implications of IPv6 Packets with Extension Headers, URL: https://www.rfc-editor.org/rfc/rfc9098

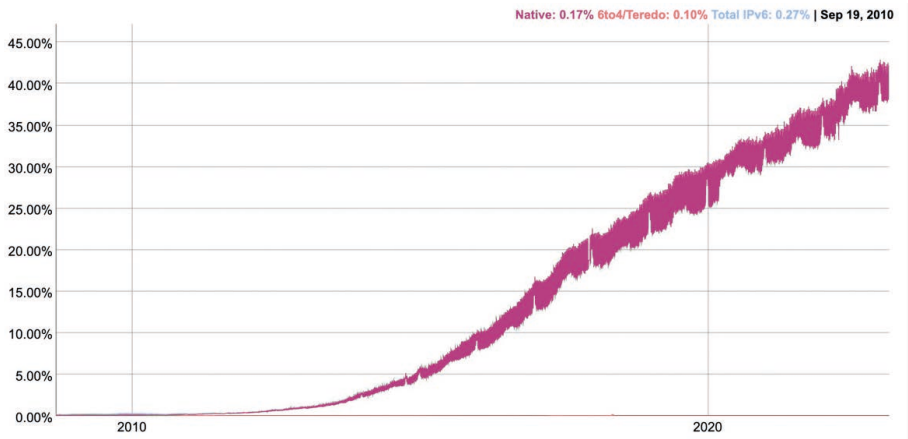

Рис. 14. Рост процента пользователей, использующих IPv6.
Источник: https://www.google.com/intl/en/ipv6/statistics.html

Поставщики контента немного отстают — процент веб-сайтов из списка 1000 наиболее популярных сайтов, доступных по IPv6, достиг почти 30%.

Однако приводить конкретные цифры в книге не имеет особого смысла — эти данные довольно быстро устареют. Вместо этого давайте рассмотрим основные аспекты готовности и соответствующие информационные ресурсы, где вы сможете найти актуальные данные.

Готовность инфраструктуры

Первый индикатор, который приходит в голову, — конечно, распределение адресного пространства IPv6. Эта цифра дает представление о максимальном числе провайдеров, внедряющих IPv6 в свою инфраструктуру, поскольку наличие адресного пространства является необходимым, но недостаточным условием его использования. Эти данные доступны на сайте NRO[26].

Более точным индикатором внедрения является процент сетей, анонсирующих адресное пространство IPv6 в Интернет. Эта информация доступна на сайте RIPE[27].

Информационные ресурсы

В конечном итоге пользователям не важно, какой протокол сетевого уровня они используют для доступа к ресурсам Интернета. Но если необходимый ресурс доступен по протоколу IPv6, а сеть провайдера и конечное оборудование (операционная система) пользователя поддерживают IPv6, то, скорее всего, для доступа будет использован именно этот протокол. Если одно из этих условий не выполняется, по-прежнему работу выполнит протокол IPv4.

[26] https://www.nro.net/statistics
[27] https://www.ripe.net/analyse/statistics/?tags=ipv6

В этом смысле доступность информационных ресурсов по протоколу IPv6 является важным индикатором готовности Интернета к новому протоколу.

Сайт World IPv6 Launch[28] до июня 2022 года отслеживал процент от 1000 самых популярных веб-сайтов, доступных по IPv6 (с июня 2022 года сайт перестал обновляться). Этот сайт также содержит список других индикаторов и измерений степени внедрения IPv6.

Фактическое использование IPv6

Наконец, для получения полной картины необходимо взглянуть на использование протокола конечными пользователями.

Одним из наиболее популярных графиков использования IPv6 пользователями Интернета является статистика Google[29]. Это неудивительно, ведь колоссальное число пользователей услуг Google и YouTube позволяет составить очень реалистичную картину.

Интересно также взглянуть на измерения исследователей APNIC, исследующих процент пользователей, которые могут использовать IPv6[30].

Сопутствующим индикатором является объем трафика, передаваемого по протоколу IPv6 в Интернете. Здесь стоит посмотреть на статистику точек обмена трафиком (например, AMS-IX[31]) или сетей распределения контента CDN (например, Akamai[32]).

Глобальная система администрирования адресного пространства

> Присвоением числовых идентификаторов также занимается Джон. Если вы разрабатываете протокол или приложение, которые предполагают использование идентификатора линка, сокета, порта, протокола или сети, пожалуйста, обратитесь к Джону за присвоением числового идентификатора.
>
> *RFC 790[33], «Присвоенные номера», 1981 г.*

[28] https://www.worldipv6launch.org/measurements
[29] https://www.google.com/intl/en/ipv6/statistics.html
[30] https://stats.labs.apnic.net/ipv6
[31] https://stats.ams-ix.net/sflow/ipv6.html
[32] https://www.akamai.com/internet-station/cyber-attacks/state-of-the-internet-report/ipv6-adoption-visualization
[33] RFC 790: ASSIGNED NUMBERS, URL: https://www.rfc-editor.org/rfc/rfc790

Итак, в рамках модели IP каждое устройство, а точнее, сетевой интерфейс каждого устройства, подключенного к Интернету, имеет уникальный IP-адрес. Для обеспечения уникальности присвоения IP-адресов необходима система учета и распределения адресных ресурсов — система администрирования адресного пространства. Начиная разговор об администрировании адресного пространства, стоит заметить, что в конце 1970-х гг. Интернет полностью относился к министерству обороны США, к тем университетам и научным центрам, которые вели работы в рамках DARPA. Более широкое подключение университетов к ARPANET и создание научно-образовательных сетей общего назначения (CSNET, а затем NSFNET) началось только в 80-х гг. прошлого столетия.

Неудивительно, что организации, обеспечивавшие координацию Интернета, осуществляли эти функции по контрактам с министерством обороны США.

Начиная с ARPANET за распределение различных цифровых идентификаторов, включая доменные имена верхнего уровня, параметры протоколов, IP-адреса и номера автономных систем, отвечала организация IANA (Internet Assigned Numbers Authority, в переводе — Администрация присвоенных номеров Интернета). Ее функции до 1999 года выполнял Институт информатики (Information Sciences Institute) Университета Южной Калифорнии (USC). Фактически же распределение адресов и номеров автономных систем было делегировано сетевому информационному центру DDN-NIC компании SRI International, который обслуживал так называемую интернет-регистратуру, или ИР (Internet Registry, IR).

Конец 80-х гг. прошлого века ознаменовался быстрым развитием компьютерных сетей, основанных на протоколе IP. И не только в США, но и за их пределами, особенно в Европе. Надо сказать, что в то время IP в Европе был своего рода гадким утенком. Существующие телефонные компании, монополисты в своей стране, продвигали сети коммутации пакетов, основанные на сетевой модели OSI (Open System Interconnection — взаимодействие открытых систем) и связанной с ней системе протоколов. Эти протоколы, большинство из которых кануло в Лету, были документированы в тщательно разработанных стандартах, основывались на семиуровневой модели (от физического до уровня приложений) и позволяли сетям и приложениям различных операторов взаимодействовать друг с другом. Все это было хорошо, за исключением того, что работа носила теоретический характер, пыталась предвидеть и решить все будущие потребности пользователей и делала это в рамках существующих телекоммуникационных моделей.

С другой стороны, практические требования существующих пользователей сетей передачи данных, в основном университетов и научно-исследовательских центров, были достаточно просты: предоставьте нам канал передачи данных за разумную цену, а с протоколами мы сами разберемся. В большинстве случаев для передачи данных использовалась более простая система TCP/IP,

хорошо зарекомендовавшая себя в научно-исследовательских сетях США. Понятно, что такие запросы не встречали радушного отклика со стороны телекоммуникационных компаний и организаций, отвечающих за стандартизацию, таких как Международная организация по стандартизации (ИСО) и Международный союз электросвязи (МСЭ).

Можно сказать, что развитие академических сетей в Европе во многом проходило под знаком борьбы демократичного и прагматичного TCP/IP с жесткой и дорогостоящей системой OSI. По своему характеру этот процесс очевидным образом соотносился с либерализацией, происходившей в Европе: окончание холодной войны и падение железного занавеса, движения за независимость и демократию. В это время создаются различные организации и сообщества для финансирования и координации развития академических сетей. Некоторые из них имеют французские названия с английскими акронимами — дань моде того времени. Например, RARE — Réseaux Associés pour la Recherche Européenne (Европейское сообщество научно-исследовательских сетей). Или RIPE — Réseaux IP Européens (Европейские IP-сети).

Интернационализация Интернета

В ответ на стремительное развитие и интернационализацию Интернета в августе 1990 года IAB (в то время Internet Activities Board — Совет, отвечающий за «определение технического направления создания стандартов и разрешение проблем в Интернете»,[34] опубликовал документ за авторством Винта Серфа (Vint Cerf) «Рекомендуемая политика распределения адресных идентификаторов Интернета» — RFC 1174[35].

Суть предлагаемых изменений:

- возможность делегирования распределения адресного пространства и номеров автономных систем другим организациям. Предполагалось, что эти организации будут утверждены Комитетом CCIRN (Coordinating Committee for Intercontinental Research Networking) — группой, созданной в 1988 году для координации глобального развития Интернета, изначально преимущественно между США и Европой. В рамках этой модели предполагалось, что ИР сохранит свою центральную функцию и продолжит осуществлять надзор над распределением адресного пространства во всем мире;
- предложение о прекращении использования статуса «подключенный» («connected») при распределении блоков адресов.

---

34 RFC 1118: The Hitchhikers Guide to the Internet,
URL: https://www.rfc-editor.org/rfc/rfc1118

35 RFC 1174: IAB Recommended Policy on Distributing Internet Identifier Assignment and IAB Recommended Policy Change to Internet "Connected" Status,
URL: https://www.rfc-editor.org/rfc/rfc1174

На последнем изменении стоит остановиться подробнее.

Суть статуса «подключенный» заключалась в разрешении обмена трафиком с сетью NSFNET — опорной научно-образовательной сетью США, которая, собственно, и являлась Интернетом. Поэтому быть «подключенным» к NSFNET означало быть подключенным к Интернету.

Проблема заключалась в том, что поддержка работы NSFNET осуществлялась на государственные деньги США. Изначально подключиться к NSFNET могли только государственные организации США или организации, спонсором которых являлось какое-либо американское государственное агентство.

Однако распространение сетевых технологий привело к тому, что сети стали создаваться самыми разнообразными организациями, включая негосударственные и коммерческие. К тому же правительство США старалось ограничить бюджетное финансирование подключения к NSFNET, соответственно, стимулируя подключенные сети к переходу в режим самоокупаемости. А наиболее очевидной стратегией в данном случае было обслуживание в том числе и коммерческих организаций.

Таким образом, в практике «подключения» появились нюансы — важным стало не то, какой тип организации к какой сети подключен, а то, какого типа трафиком обмениваются эти сети. Например, считалось недопустимым использование бюджетных сетей для обмена коммерческим трафиком, однако обмен трафиком между коммерческой организацией и университетом в рамках научно-исследовательского проекта был вполне возможен.

Единовременно присуждаемый статус более не являлся работающим критерием, и требовался более тонкий контроль за соблюдением так называемых правил, или политики допустимого использования (AUP — Acceptable Use Policy). Вместо бинарного статуса IAB предлагал сбор информации о политике использования подключенных сетей. Контроль за соответствием политики маршрутизации политике использования оставался за NSFNET, которая вела соответствующую базу данных (Policy Routing Database, PRDB, позже трансформированную в регистратуру маршрутизации RADB).

Краткая история политики распределения адресных ресурсов в Европейском регионе

В том же августе 1990 года на рассмотрение участников 6-го совещания RIPE было вынесено 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;[35]
- информационное обслуживание сетей;
- административная поддержка RIPE.

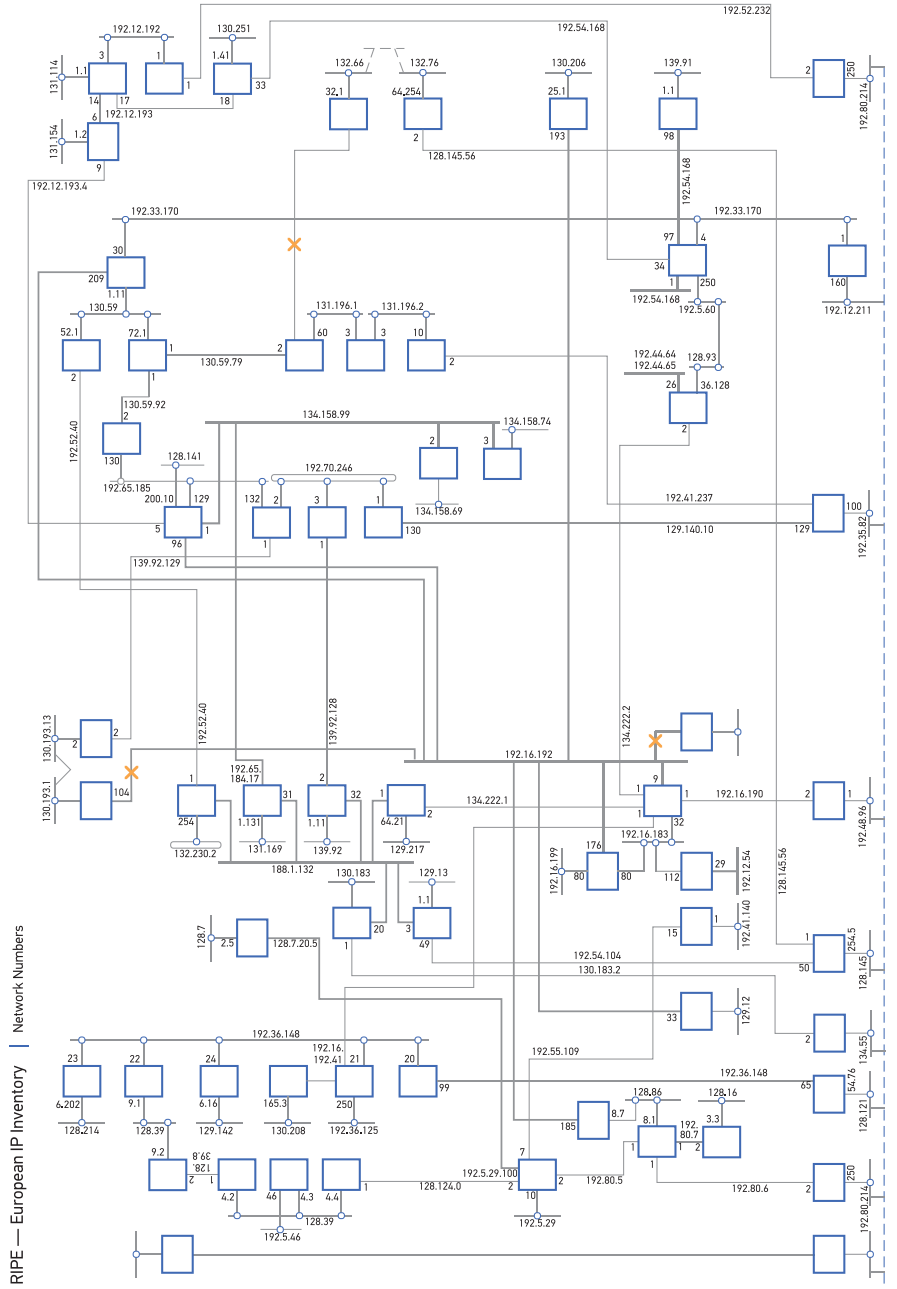

Рис. 15. Европейские IP-сети в 1990 году

Источник: ftp://ftp.ripe.net/ripe/docs/ripe-026.ps)

Назначение Даниела Карренберга (Daniel Karrenberg) генеральным директором RIPE NCC, а также размещение центра в нидерландском Национальном институте ядерной физики и физики высоких энергий NIKHEF было утверждено в январе 1992 года советом RARE — организации, которая обеспечивала юридическую платформу для нового координационного центра. В апреле того же года RIPE NCC был формально представлен участникам XII совещания RIPE. Центр арендовал две комнаты в NIKHEF, а его штат насчитывал три человека.

Одной из задач RIPE NCC являлось создание «делегированной» интернет-регистратуры в рамках концепции, предложенной в RFC 1174.[35] Уже в сентябре 1992 г. RIPE NCC начал обслуживать запросы на получение адресного пространства от европейских организаций.

Двумя месяцами раньше, в июле, был опубликован документ (RIPE NCC Internet Numbers Registration Procedures, ripe-065[36]), который можно считать первой политикой распределения ресурсов. Данная процедура регистрации адресов содержала ряд требований:

- адресное пространство выделялось только сервис-провайдерам, которые, в свою очередь, распределяли его индивидуальным организациям;
- распределенное адресное пространство подлежало регистрации в базе данных RIPE;
- дополнительное обоснование требовалось для запроса сети класса B (это было время «классовых» сетей, хотя уже применялась концепция супернетов — объединение нескольких последовательных адресных блоков класса C в сеть большего размера, — для распределения адресных ресурсов, технология CIDR еще не была внедрена);
- для сетей класса C устанавливалось требование поддержки супернетов, а именно — требование резервирования сервис-провайдерами соседних блоков класса C с целью возможности их последующего объединения.

Процедура являлась достаточно простой и неформальной. Немногим позже она была слегка формализована: появилась форма запроса, в которой, помимо контактных данных, заявитель предоставлял сведения о размере сети и перспективах ее роста на ближайшие два года. Обсуждение этих документов происходило в рамках образованной рабочей группы Local IR.

В то время разработка системы и политики распределения адресных ресурсов происходила в рамках IETF. Помимо упомянутого основополагающего документа RFC 1174[35], в мае 1993 года вышел более полный документ «Руководство по управлению адресным пространством» — RFC 1466.[37] Этот документ, который активно обсуждался и был согласован с RIPE, явился основой политики распределения адресных ресурсов в Европе.

---

[36] https://www.ripe.net/publications/docs/ripe-065

[37] RFC 1466: Guidelines for Management of IP Address Space, URL: https://www.rfc-editor.org/rfc/rfc1466

Совместно с системой CIDR эти документы обозначили становление новой иерархической системы распределения адресного пространства, верхний уровень которой базировался на геополитическом принципе, а последующие — на системе взаимоотношений провайдер-клиент. Последняя предусматривала, что адресное пространство, полученное сервис-провайдером от региональной интернет-регистратуры (РИР), будет далее иерархически распределено этим провайдером его клиентам и т.д. Последствием внедрения такой системы явилась зависимость адресации сетей клиентов от конкретного провайдера и требование переадресации в случае, когда сеть меняет своего провайдера.

Существовала, правда, возможность получить непосредственно от РИР адресное пространство, которое являлось независимым от провайдера. Такие ресурсы так и назывались — независимые от провайдера (Provider Independent, PI). Все остальные получили название «агрегируемые» (Provider Aggregatable, PA). Независимые ресурсы являлись более «дорогостоящими» для Интернета, поскольку не могли быть агрегированы в большие блоки и тем самым оказывали большую нагрузку на систему маршрутизации. Надо сказать, что в то время ресурсы маршрутизаторов были достаточно ограничены и рост маршрутизационных таблиц представлял серьезную проблему. Было даже предложено остановить выдачу ресурсов PI, однако RIPE выступил против, опасаясь последствий в виде усиления регулирующих функций RIPE NCC. В ответ было решено усилить разъяснительную работу среди запрашивающих ресурсы PI, акцентируя внимание на то, что данные ресурсы могут иметь проблемы с глобальной маршрутизацией.

Летом 1995 года концепция «агрегируемых» (Provider Aggregatable) и «независимых» от провайдера (Provider Independent) адресных ресурсов была включена в политику распределения адресных ресурсов и документирована в RIPE-127[38].

В это же время перед Интернетом стояла еще одна проблема — назревающая нехватка адресного пространства. Хотя внедрение CIDR отсрочило опустошение пула свободных адресов, внимание общественности было привлечено к более бережливому распределению конечного ресурса. Надо отметить, что решение задачи сохранения адресных ресурсов усугубляет проблему роста таблиц маршрутизации — и наоборот. Дело в том, что для сохранения адресного пространства желательно выделять как можно меньшие блоки адресов, минимизируя резервирование, в то время как для оптимальной маршрутизации важно, чтобы адресные блоки сервис-провайдера были максимально агрегируемы. Вопрос верного баланса между этими противоречащими друг другу целями впервые появился на повестке дня XXII конференции RIPE (январь 1996 года) и с тех пор является ключевым в обсуждении различных правил и параметров распределения. Осенью 1996 года были опубликованы два документа. Один, традиционно подготовленный в рамках IETF и опубликованный под номером

[38] ftp://ftp.ripe.net/ripe/docs/ripe-127.txt

RFC 2050,[39] назывался «Руководство по распределению IP-адресов интернет-регистратурой». По существу, данный RFC документировал существовавшую до этого времени практику распределения, которая являлась основой политик RIPE. Второй документ назывался «Политика и процедуры европейской интернет-регистратуры» (European Internet Registry Policies and Procedures, RIPE-140[40]) и являлся обобщением принципов, правил и процедур, связанных с распределением адресного пространства RIPE NCC. Этот документ стал плодом многомесячного обсуждения в рамках рабочей группы Local IR и обозначил образование независимой региональной политики RIPE. С этого момента процесс разработки политик, связанных с распределением адресных ресурсов, происходит в списках рассылки рабочей группы Local IR.

Принципы распределения адресного пространства

Документ RIPE-140 представил основные принципы распределения адресного пространства.

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

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.

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

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

К этим принципам добавился принцип «Справедливости» — все политики и практики, связанные с использованием пространства публичного оповещения, должны справедливо и равноправно применяться ко всем существующим и потенциальным членам интернет-сообщества, независимо от их местонахождения, национальности, размера или любого другого фактора.

Принцип «Сохранения» для адресного пространства IPv4 особого смысла не имеет и поэтому из текущей политики[41] исключен.

---

[39] RFC 2050: Internet Registry IP Allocation Guidelines, URL: https://www.rfc-editor.org/rfc/rfc2050

[40] ftp://ftp.ripe.net/ripe/docs/ripe-140.txt

[41] RIPE-733, URL: https://www.ripe.net/publications/docs/ripe-733

Хотя для адресного пространства IPv6 принцип «Сохранения» кажется менее важным, чем в свое время в отношении IPv4, политика распределения[42] стремится обеспечить оптимальный баланс между противоречащими принципами агрегируемости и сохранения. По существу этих параметров три:

- **Минимальный размер распределяемого пространства**. Чем больше минимальный размер, тем выше вероятность неиспользуемых запасов. Для адресного пространства IPv6 этот параметр равен /32.
- **Временной интервал планирования** 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. Для адресов IPv6 этот параметр определен менее четко, чем в случае IPv4 (для IPv4 он был равен 12 месяцам), в силу значительно большего размера адресного пространства, получаемого изначально. В принципе, горизонт планирования составляет два года.
- **Процент использования распределенных ресурсов (утилизация)**. Для оценки утилизации адресного пространства IPv6 используется коэффициент HD-ratio (Host-Density Ratio), документированный в RFC 3194[43]. Для получения последующего блока адресов провайдер должен продемонстрировать утилизацию HD-ratio не менее 0,94.

Современная система

Вслед за созданием координационного центра RIPE NCC и признанием его в качестве региональной регистратуры в 1994 года последовало официальное признание IANA второй регистратуры — APNIC, отвечавшей за распределение адресного пространства и номеров автономных систем в Азиатско-Тихоокеанском регионе.

Создание третьей региональной интернет-регистратуры, ARIN, было связано с процессом приватизации Интернета в США. Изначально интернет-регистратура обслуживалась исследовательским институтом SRI International. Учитывая стремительное развитие Интернета, в том числе и в коммерческом секторе, правительство США и NSF приняли решение об изменении существующей структуры ИР, финансируемой министерством обороны. Начиная с 1991 года обслуживание ИР, которая теперь стала называться InterNIC и занималась в том числе и распределением доменных имен, переходит к небольшой (в то время) компании Network Solutions Incorporated (NSI).

Региональные регистратуры RIPE NCC и APNIC эффективно обслуживали соответствующие сообщества операторов в соответствии с ими же разработанными политиками. Этот успех явился одним из факторов разделения функции распределения доменных имен и создания в США отдельной регистратуры для номерных

[42] RIPE-738, URL: https://www.ripe.net/publications/docs/ripe-738

[43] RFC 3194: The Host-Density Ratio for Address Assignment Efficiency: An update on the H ratio, URL: https://www.rfc-editor.org/rfc/rfc3194

ресурсов. В декабре 1997 года была создана Американская регистратура интернет-номеров ARIN.

В результате мир был разделен на три сферы обслуживания. Европейские страны, страны бывшего СССР, Ближнего Востока, а также Северной Африки обслуживались RIPE NCC; Азиатско-Тихоокеанский регион обслуживался APNIC, офис которого к тому времени переместился из Японии в Австралию, г. Брисбен. Наконец регистратура ARIN обслуживала Северную Америку и все остальные регионы — страны Латинской Америки, Карибского бассейна и южной части Африки.

Когда в начале XXI века стало очевидно, что нужна отдельная организация LACNIC (Латиноамериканская и Карибская региональная регистратура) для обслуживания латиноамериканских сетей, Интернет уже давно вырос из научно-образовательной сети США и стал глобальной сетью, обладающей громадным экономическим и социальным потенциалом. Формализованы были процессы принятия важных решений, в них участвовало международное сообщество. Процесс создания новых региональных регистратур был установлен в документе ICP-2 «Критерии создания новых региональных интернет-регистратур»[44], созданном в процессе консультаций между RIPE NCC, APNIC, ARIN и ICANN, а также соответствующими сообществами. Одними из важнейших критериев являлись нейтралитет, техническая экспертиза и широкая поддержка регионального сообщества сетевых операторов.

После годового пробного срока 7 ноября 2002 года IANA официально объявила о признании LACNIC четвертой РИР. Часть зоны обслуживания ARIN отошла к новой РИР.

За LACNIC последовало создание AfriNIC (в апреле 2005 года) под чью ответственность попал весь Африканский континент, до этого обслуживаемый ARIN и RIPE NCC.

В октябре 2003 года APNIC, ARIN, LACNIC и RIPE NCC заключили соглашение об образовании Организации номерных ресурсов[45]. В 2005 году к ним присоединился AfriNIC.

Задачи NRO — обеспечение единого интерфейса взаимодействия системы РИР с внешним миром, а также координация различных совместных проектов регистратур.

Руководством деятельности NRO занимается исполнительный комитет, состоящий из пяти руководителей РИР (формально члены исполкома назначаются советом каждого РИРа).

NRO участвует в разработке политик через Номерной комитет (NRO Number Council) — но, по существу, это независимая от NRO структура, тождественная Совету поддерживающей организации по адресации ICANN (Address Supporting

[44] https://www.icann.org/resources/pages/new-rirs-criteria-2012-02-25-en
[45] Number Resource Organization, NRO, URL: http://www.nro.net

Organization, ASO). Каждое из региональных сообществ РИР выбирает двух членов для включения в состав Номерного комитета (и, соответственно, в Совет ASO).

Процесс разработки политики распределения номерных ресурсов Интернета

Хотя RIPE-1ao и определил основные принципы и правила распределения номерных ресурсов (адресных блоков IPv4 и номеров автономных систем), жизнь вносила свои коррективы, возникали новые требования. До некоторого времени процесс разработки новых правил распределения адресов носил достаточно неформальный характер. Работа над определенной политикой была открытой, и любой желающий мог принять в ней участие. Как правило, процесс начинался с подготовки проекта в свободном формате. Он описывал проблему — например, отсутствие правил для определенной ситуации — и выдвигал предложение по ее решению.

В результате последующего обсуждения предложение либо браковалось (например, из-за отсутствия интереса к представленной проблеме), либо дорабатывалось и, возможно, принималось.

Хотя процесс не был формализован, он основывался на трех принципах:

- **Открытость.** Любой желающий может предложить политику и участвовать в ее обсуждении. При этом в расчет принимается аргументация, а не ранг участника или его работодателя.
- **Прозрачность.** Обсуждения и результаты обсуждений документированы и свободно доступны каждому.
- **Консенсус.** Решения принимаются на основе консенсуса.

Обсуждение предложения в основном происходило в соответствующих списках рассылки, а конференции RIPE использовались для более интерактивного обмена мнениями. Зачастую предварительное «зондирование почвы» на предмет интереса к предполагаемой проблеме происходило также на конференции RIPE.

Такая система прекрасно работала, пока сообщество RIPE было небольшим и однородным. Однако по мере роста сообщества усиливались требования по структуризации и формализации процесса. Например, не все предложения и обсуждения носят одинаковый характер. Часть из них является просто обсуждением рекомендаций и технических практик. Другие же, наоборот, ставят своей целью разработку требований или запроса к RIPE NCC. Наконец, ряд дискуссий в действительности затрагивают принципиальные моменты, являющиеся частью существующей или новой политики RIPE. Не все требуют одинакового «обхождения» и форумов для обсуждения.

Еще одним недостатком существовавшего процесса являлось отсутствие четких сроков подготовки и обсуждения предложения. В результате процесс мог либо затянуться на неопределенное время, либо закончиться неожиданно быстро для тех, кто не особенно внимательно следил за его развитием.

Хотя все обсуждения и решения документировались (архивы списков рассылки, протоколы совещаний на конференциях RIPE, документы RIPE), отсутствовала документация самого процесса и его фаз. В результате отслеживание разработки политик порой требовало значительных усилий.

В сентябре 2005 года был опубликован документ «Процесс разработки политики в RIPE» (Policy Development Process in RIPE) под авторством Роба Блокзайла (Rob Blokzijl), описывающий основные принципы и элементы этого процесса. С тех пор документ прошел через несколько ревизий, но его основа сохранилась прежней.[46]

По существу, этот документ — сжатая инструкция для разработчика политики. За отправную точку взят следующий принцип: единственным требованием для участия в процессе разработки политики, или ПРП — от изначальной идеи, обсуждения предложения до согласованной политики — является доступ к электронной почте и подписка на список рассылки соответствующей рабочей группы, в рамках которой предполагается обсуждение предложения. Для участия необязательно быть членом RIPE NCC или участником конференций RIPE. Однако важно иметь предложение, которое решало бы насущную проблему.

Таким предложением может быть изменение или дополнение в существующую политику либо совершенно новая политика RIPE. Если такое предложение имеется, необходимо соответствующим образом оформить его (структура проекта политики приведена в документе). Проект политики обычно представляется через председателя соответствующей рабочей группы. В случае, когда неочевидно, к какой рабочей группе относится предложение, проект можно представить через председателя RIPE по адресу policyproposal@ripe.net. Кстати, RIPE NCC может оказать помощь в подготовке проекта.

После подачи проект должен пройти три стадии до возможного утверждения в качестве новой политики. Диаграмма на рис. 16 показывает эти стадии на временной оси.

Сначала предложение проходит стадию **обсуждения** (Discussion phase). Эта фаза начинается с анонсирования нового предложения. Для этого используется список policy-announce@ripe.net. Тогда же сообщается, в какой рабочей группе будет происходить обсуждение. Председатель рабочей группы устанавливает продолжительность периода обсуждения, это как минимум четыре недели. По окончании данной стадии предложение может перейти в следующую стадию, отправлено на доработку с повторным обсуждением или вообще исключено. Это зависит от характера замечаний и комментариев и решается по согласованию между подателем предложения и председателем рабочей группы.

Если проект переходит в следующую фазу — **рецензирования** (Review phase), — то в течение четырех недель необходимо подготовить начальную версию документа RIPE — официального способа публикации утвержденных политик.

[46] Текущая версия — https://www.ripe.net/publications/docs/ripe-642

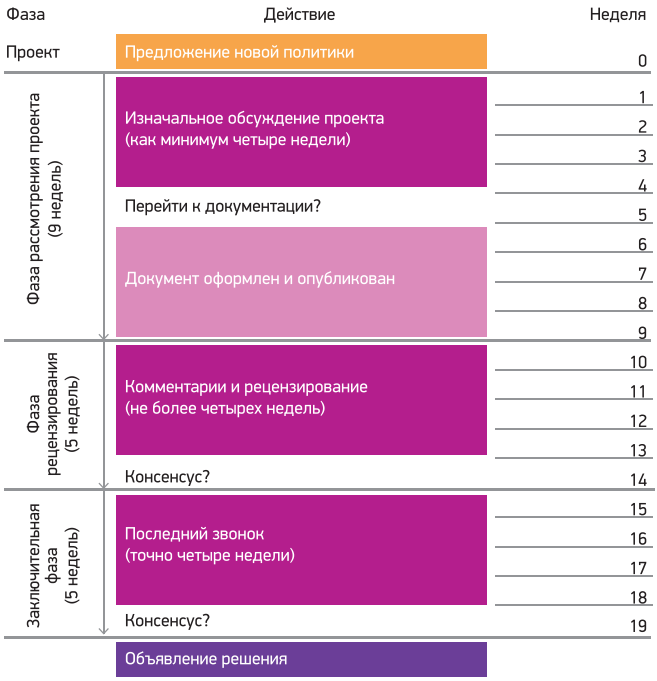

Рис. 16. Схема процесса разработки политики (ПРП) RIPE.
Источник: https://www.ripe.net/publications/docs/ripe-642

В стадии рецензирования работа идет над фактическим текстом политики — таким, каким он появится в окончательном варианте. Максимально отведенное время — также четыре недели. По прошествии этого времени председатель решает, был ли достигнут консенсус в отношении проекта. В случае положительного решения проект переходит в **заключительную** стадию (Conclusion phase). В противном случае председатель может полностью исключить проект, отправить на доработку предложения или текст проекта (в зависимости от этого процесс начинается либо со стадии обсуждения, либо рецензирования).

Заключительная стадия начинается с объявления «последнего звонка» (Last Call) в списках рабочей группы и policy-announce@ripe.net. В течение последующих четырех недель любой член сообщества может направить свои комментарии.

Цель — дать возможность противникам политики обозначить свою позицию, если они по каким-либо причинам не сделали это на предыдущих стадиях. По прошествии четырех недель председатели всех рабочих групп совместно решают, был ли достигнут консенсус.

В случае положительного решения объявляется новая действующая политика. Если, по мнению председателей, консенсус не был достигнут, председатели могут исключить проект или направить его на повторное обсуждение.

Все текущие предложения представлены на сайте RIPE[47]. Там же можно узнать, в какой стадии находится то или иное предложение, дату окончания дискуссий и рабочую группу, в которой ведется обсуждение.

В ПРП существует несколько моментов, когда принимается решение о дальнейшей судьбе предложения. Возможно, что участники обсуждения не согласны с решением председателя. Для таких случаев предусмотрена специальная процедура разрешения споров. Надо отметить, что к рассмотрению принимаются претензии только по решению о достижении или отсутствии консенсуса. Разногласия относительно самого предложения, его технических и прочих качеств должны разрешаться в ходе самого обсуждения.

Возможен вариант, при котором — с точки зрения участника работы над предложением — его взгляды не были приняты к рассмотрению в стадии обсуждения. Или решение о достижении консенсуса по окончании стадии рецензирования кажется неправомерным. В таком случае участник должен попытаться разрешить этот вопрос с председателем. Если это невозможно, задача разрешения спора решается председателями всех рабочих групп совместно путем голосования. Председатель, вовлеченный в спор, от голосования воздерживается. Это решение председателей является окончательным.

Другой пример спорной ситуации — несогласие с решением о достижении консенсуса в заключительной стадии, которое совместно приняли председатели всех рабочих групп. В этом случае высшей инстанцией разрешения спора является председатель RIPE. Опять же, его решение в этом споре является окончательным.

Решение об исключении не означает, что предложение похоронено навсегда. В любой момент процесс может быть начат заново — бывает, что новую жизнь обретает даже то же самое предложение. Например, если обнаружились новые убедительные аргументы в его пользу или возникло новое предложение, основанное на изначальной идее.

Наиболее важные политики

За годы существования RIPE сообществом были разработаны десятки различных политик, начиная с общих политик распределения номерных ресурсов (адресов и номеров автономных систем) до специальных случаев и рекомендаций. Дюжина документов регламентирует современный процесс распределения номерных ресурсов в RIPE. Все эти документы доступны на сайте RIPE[48].

Текущие политики распределения номерных ресурсов можно условно разделить на несколько категорий: глобальные, общие и специальные региональные политики.

[47] https://www.ripe.net/participate/policies/current-proposals/current-policy-proposals
[48] https://www.ripe.net/publications/docs/ripe-policies/ripe-policies

Глобальные политики

Пример глобальных политик — распределение блоков IPv4, IPv6 и номеров автономных систем от IANA региональным интернет-регистратурам. К глобальным относится политика, достигшая консенсуса во всех регионах (РИРах) и ICANN и требующая участия IANA или какой-либо внешней организации, связанной с ICANN, для ее осуществления. Процесс обсуждения и достижения консенсуса происходит во всех регионах более или менее независимо, следуя своим региональным правилам. Например, в RIPE обсуждение глобальной политики должно следовать ПРП. По достижении консенсуса во всех регионах проект политики рассматривается Советом ASO — комитета ICANN по вопросам номерных ресурсов. Если, по мнению Совета ASO, результирующий текст адекватно представляет обсуждавшийся проект, процесс достижения консенсуса не был нарушен и мнения основных заинтересованных сторон были учтены, текст направляется в Совет ICANN для ратификации. При положительном решении новая глобальная политика вступает в действие. Схема процесса показана на рис. 17, а его полное описание можно найти на сайте NRO[49].

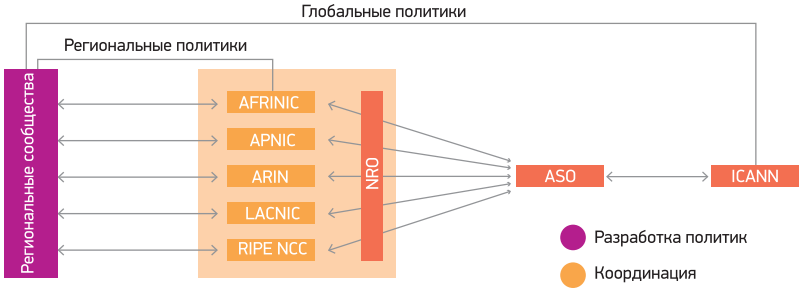

Рис. 17. Процесс разработки и принятия глобальной политики.
Источник: http://www.nro.net/policies/global-policies-development-process

Перечислим текущие глобальные политики.

В настоящий момент существуют три глобальные политики, определяющие распределение адресных ресурсов от IANA региональным интернет-регистратурам:

- Политика IANA распределения блоков номеров автономных систем региональным интернет-регистратурам[50].
- Политика IANA распределения адресных блоков IPv6 региональным интернет-регистратурам[51].

---

49 https://www.nro.net/global-policy-development-process

50 Internet Assigned Numbers Authority (IANA) Policy for Allocation of ASN Blocks to Regional Internet Registries, https://www.icann.org/resources/pages/global-policy-asn-blocks-2010-09-21-en

51 Internet Assigned Numbers Authority (IANA) Policy for Allocation of IPv6 Blocks to Regional Internet Registries, https://www.icann.org/resources/pages/allocation-ipv6-rirs-2012-02-25-en

- Глобальная политика, определяющая механизмы распределения оставшихся адресов IPv4[52].

Последняя политика заменила глобальную политику распределения адресного пространства IPv4 после опустошения пула свободных адресов IPv4 IANA.

Общие региональные политики распределения ИР

Эти политики были утверждены сообществом RIPE и определяют принципы и правила, которым следует RIPE NCC при распределении ресурсов локальным интернет-регистратурам — ЛИРам. К этим политикам относятся:

- политика распределения адресного пространства IPv6[53];
- политика распределения адресного пространства IPv4[54];
- политика распределения номеров автономных систем[55].

Заметим, что подобные политики существуют во всех регионах. Ежеквартально NRO подготавливает обзорный документ, сравнивающий политики и практики РИРов. Эти документы доступны на сайте NRO[56].

Специальные региональные политики распределения ИР

Эти политики охватывают специальные случаи. Например:

- распределение номерных ресурсов для использования самим RIPE NCC — Allocating/Assigning Resources to the RIPE NCC; данная политика определяет, как сам RIPE NCC может запросить и, возможно, получить ресурсы от RIPE NCC;
- требования к договорным отношениям для держателей ресурсов, не зависимых от провайдера — Contractual Requirements for Provider Independent Resource Holders in the RIPE NCC Service Region; данная политика требует обязательного наличия договорных отношений между держателями таких ресурсов и «регистратором» этих ресурсов; в качестве регистратора может выступать либо ЛИР (в большинстве случаев это ЛИР, через которую и были получены данные ресурсы), либо RIPE NCC;
- специальный случай получения ресурсов IPv6 для точек обмена трафиком — IPv6 Address Space Policy for Internet Exchange Points;
- специальный случай получения ресурсов IPv6 для серверов корневой зоны DNS — IPv6 Addresses for Internet Root Servers in the RIPE Region.

---

52 Global Policy for Post Exhaustion IPv4 Allocation Mechanisms by the IANA, https://www.icann.org/resources/pages/allocation-ipv4-post-exhaustion-2012-05-08-en

53 IPv6 Address Allocation and Assignment Policy, https://www.ripe.net/publications/docs/ripe-738

54 IPv4 Address Allocation and Assignment Policies for the RIPE NCC Service Region, https://www.ripe.net/publications/docs/ripe-733

55 Autonomous System (AS) Number Assignment Policies, https://www.ripe.net/publications/docs/ripe-679

56 https://www.ripe.net/publications/docs/ripe-804

Некоторые общие политики описывают и специальные случаи. Например, выдача адресных блоков операторам доменов верхнего уровня и ENUM в случае использования технологии аникаст (anycast) рассматривается в политике распределения адресного пространства IPv6.

Рекомендации, процедуры и формы

В то время как политики определяют основополагающие принципы и правила, их конкретное воплощение отражается в разработанных RIPE NCC процедурах и формах. Эти документы можно найти на сайте RIPE[57] в разделе «Request Forms & Supporting Notes».

Заключение

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 и 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 — но все эти впечатляющие изменения происходят на уровнях ниже и выше IP. На самом же уровне IP время как будто остановилось. 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.

Времени до того момента, когда переход на IPv6 станет не опцией, а единственным возможным путем развития Интернета, к сожалению, остается все меньше.

[57] https://www.ripe.net/publications/docs/ripe-documents


ЗАПИСЬ РЕСУРСА CCNSO
РЕГИСТРАТОР
IDNA2008
РЕГИСТРАТОР
АВТОРИТЕТНЫЙ СЕРВЕР
RFC1034
POWERDNS
УЗЕЛ
ЗАПРОС
.GOV
.MIL
.NET
.EDU
.ORG
.COM
NSD
POWERDNS
IDNA2008
DNS
BIND
IDNA2008
RFC5155
РЕЗОЛВЕР
RFC4034
GTLD
PTR
DNSSEC
УСТОЙЧИВОСТЬ
ЗОНА
RFC1034
ЗАПИСЬ РЕСУРСА
RFC4033
EDNS0
BIND
RFC1035
СЕРВЕР ИМЕН
ЗАПРОС
UNBOUND
ИНТЕРНАЦИОНАЛИЗАЦИЯ
БЕЗОПАСНОСТЬ
РЕФЛЕКТОРНЫЕ АТАКИ
IDNA2008
ДОМЕННОЕ ИМЯ
A
РЕГИСТРАТОР
ОТВЕТ
РЕЗОЛВЕР
IDNA2008
DIG
POWERDNS
AAAA
БЕЗОПАСНОСТЬ
IDNA2008
DDOS
АВТОРИТЕТНЫЙ СЕРВЕР
RFC4033
MSK-IX
RFC1034
ЗАПРОС
ИНТЕРНАЦИОНАЛИЗАЦИЯ
POWERDNS
PTR
RFC1035
РЕГИСТРАТУРА
RFC4035
POWERDNS
АНИКАСТ
СЕРВЕР ИМЕН
RFC4035
IDNA2008
АНИКАСТ
КООРДИНАЦИЯ
СЕРВЕР ИМЕН
IANA
КЛЮЧ
ICANN
БЕЗОПАСНОСТЬ
РЕФЛЕКТОРНЫЕ АТАКИ
RFC4035
РЕФЛЕКТОРНЫЕ АТАКИ
ОТВЕТ
EDNS0
DDOS
RFC5155
РЕГИСТРАТОР
CCNSO
ВТОРИЧНЫЙ СЕРВЕР
RFC4035
УСТОЙЧИВОСТЬ
GNSO
РЕЗОЛВЕР
РАСШИРЕНИЯ
GNSO
IDNA2008
AAAA
РАСШИРЕНИЯ
RFC4033
UNBOUND
WHOIS
БЕЗОПАСНОСТЬ
IANA
RFC4033
КООРДИНАЦИЯ
БЕЗОПАСНОСТЬ
ДОМЕННОЕ ИМЯ
DDOS
ЗАПРОС
ИНТЕРНАЦИОНАЛИЗАЦИЯ
УСТОЙЧИВОСТЬ
КОРНЕВОЙ СЕРВЕР
POWERDNS

Глава 2

Глобальная система имен

> Наконец-то у нас есть официальный список имен хостов. Теперь самое время положить конец той абсурдной ситуации, когда каждый узел сети вынужден обслуживать собственный список хостов — как правило, устаревший и отличающийся от других — для работы операционной системы или пользовательских программ.
>
> *RFC 606[1], декабрь 1973 г.*

Глобальная система доменных имен — Domain Name System, или DNS — 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. С 1987 года, когда была завершена работа над основной спецификацией сегодняшней DNS (RFC 1034[2] и RFC 1035[3]), до настоящего времени

[1] RFC 606: Host Names On-line, URL: https://www.rfc-editor.org/rfc/rfc606

[2] RFC 1034: Domain Names Concepts and Facilities, URL: https://www.rfc-editor.org/rfc/rfc1034

[3] RFC 1035: Domain Names Implementation and Specification, URL: https://www.rfc-editor.org/rfc/rfc1035

было выпущено более 500 RFC, определяющих дополнительные функции системы или так или иначе связанных с ее работой. Но DNS также и глобальная распределенная база данных, хранящая сотни миллионов имен и связанных с ними ресурсов. Развиваясь вместе с самим Интернетом, DNS сегодня обслуживается более чем 16 миллионами серверов, обрабатывая несколько десятков миллионов запросов в секунду.

Нормальная работа сети Интернет немыслима без правильно функционирующей DNS. Всякий раз, когда мы набираем имя веб-сайта или отправляем электронную почту, эта система берет на себя задачу трансляции имени в цифровой адрес протокола IP, необходимый для осуществления связи между компьютерами в сети. Строго говоря, для работы IP-протокола и, соответственно, для обмена данными между компьютерами DNS не требуется. Но сегодня зависимость от этой системы настолько велика, что существенный сбой в ее работе фактически приведет к остановке самого Интернета.

Система DNS является иерархической и распределенной. Не существует единой базы данных, хранящей информацию обо всех именах, соответствующих им IP-адресах и других записях. Напротив, DNS — это миллионы баз данных, или как их чаще называют — «зон», каждая из которых содержит информацию о конкретном домене. Как правило, каждая зона обслуживается двумя или более серверами, отвечающими на запросы клиентов.

Такая архитектура DNS позволяет, во-первых, обеспечить уникальность имен, а во-вторых, распределить нагрузку и ответственность за работу системы между администраторами отдельных доменов. Каждая зона независимо обслуживается администратором зоны, отвечающим за ее содержимое, производительность и бесперебойную работу. Эта модель обеспечила долголетие системы и ее эволюционное развитие уже на протяжении более чем трех десятилетий.

Краткая история DNS

Идея использовать имена, имеющие смысловое значение, вместо числовых идентификаторов родилась почти одновременно с ARPANET. Во-первых, имя SRI-ARC или UCLA-CCn было легче запомнить, чем 002.2 или 101.65[4]. Во-вторых, в приложениях стало возможным использовать постоянные имена, не заботясь об изменениях инфраструктуры и адресов компьютеров.

Однако в то время DNS еще не существовала. Для обеспечения соответствия между именами и адресами компьютеров использовался специальный, хранимый на каждом компьютере файл, впоследствии получивший имя hosts.txt. Хотя изначально список был небольшим, насчитывавшим к концу 1973 года всего

[4] RFC 597: Host Status, URL: https://www.rfc-editor.org/rfc/rfc597

80 хостов[5], во избежание ошибок в списке требовалась координация, а именно централизация публикации и обслуживания списка. Эта функция была возложена на сетевой информационный центр в SRI, о котором мы уже говорили в первой главе. Для публикации и доступа к списку использовался протокол FTP.

К концу 1981 года, однако, число хостов перевалило за 500, и обслуживать неструктурированные имена становилось все более проблематичным. Да и список, локальная копия которого присутствовала на каждом компьютере, стал довольно громоздким.

Идея использования иерархических доменных имен родилась при решении проблемы уникальности почтовых адресов в системе доставки электронной почты в растущем Интернете. В феврале 1982 года для обсуждения возможного решения этой проблемы было организовано совещание, основные моменты которого задокументированы в RFC 805[6]. В частности, документ указывает, что «используемый в настоящее время идентификатор «user@host» должен быть расширен в «user@host.domain», где domain может представлять собой иерархию доменов».

В этом же документе можно найти наметки будущей системы DNS — систему серверов имен, в ответ на запрос возвращающих или адрес компьютера получателя почты, или же адрес сервера имен домена получателя почты (в сегодняшнем жаргоне DNS — referral, или перенаправление). Эти идеи приобрели более конкретные очертания в документе RFC 819[7], опубликованном через полгода. В частности, в документе рассматривались различные типы серверов имен, иерархия имен, а также был определен первый домен верхнего уровня — ARPA, объединявший организации, подключенные к Интернету в рамках проекта DARPA[8]. В октябре того же 1982 года появляется документ RFC 830[9] под авторством сотрудника SRI Зо-Синг Су (Zaw-Sing Su) «Распределенная система для реализации услуги имен Интернета». В нем были приведены полная кон-

---

5 Там же

6 RFC 805: Computer Mail Meeting Notes,
URL: https://www.rfc-editor.org/rfc/rfc805

7 RFC 819: The Domain Naming Convention for Internet User Applications,
URL: https://www.rfc-editor.org/rfc/rfc819

8 В 1987 г. в домене ARPA появился поддомен IN-ADDR.ARPA, который используется для обратной трансляции — трансляции адресов в доменные имена. В 2000 г. IAB опубликовал заявление, в котором предлагается использовать ARPA в качестве основного инфраструктурного домена (ранее предполагалось для этого использовать INT, но он больше подходил для международных организаций). Также предлагалось изменить аббревиатуру: ARPA теперь расшифровывалась как Address and Routing Parameters Area (область параметров маршрутизации и адресации).
URL: http://on-infrastructure-domain-and-subdomains-may-2000

9 RFC 830: A Distributed System for Internet Name Service,
URL: https://www.rfc-editor.org/rfc/rfc830

цепция и архитектура DNS, во многом оставшиеся неизменными в сегодняшней системе.

С этого момента развитие DNS происходит стремительно. Уже в ноябре 1982 года Пол Мокапетрис (Paul Mockapetris) публикует RFC 882[10] «Доменные имена — концепции и услуги» и RFC 883[11] «Доменные имена — исполнение и спецификация», которые явились первыми стандартами системы доменных имен.

В 1987-м изначальные спецификации RFC 882 и RFC 883 были дополнены с учетом опыта разработки приложений и внедрения. Обновленные стандарты, RFC 1034[12] и RFC 1035[13], и по сей день остаются основными спецификациями DNS.

Архитектура и работа DNS

Как было сказано выше, система DNS является иерархической и распределенной. Не существует единой базы данных, хранящей информацию обо всех именах и соответствующих им IP-адресах и других записях. Каждый домен, определяющий собственные имена и поддомены, является отдельной базой данных, или в терминах самой DNS — зоной. Иерархию DNS можно увидеть в доменном имени. Например, имя www.example.com. состоит из трех частей, разделенных точками. Точнее, четырех, поскольку, формально говоря, полное доменное имя всегда заканчивается точкой, обозначающей так называемый корневой домен, или корневую зону DNS. Итак:

. Корневая зона, содержащая информацию обо всех поддоменах: net, com, org, ru, su и т.д. Точнее, там содержится информация о серверах, обслуживающих эти домены.

com Домен com, содержащий информацию обо всех поддоменах, зарегистрированных в нем. В частности, о поддомене example. Опять же, эта информация включает адреса серверов, у которых можно получить дополнительную информацию о содержимом поддоменов.

---

10 RFC 882: Domain Names — Concepts and Facilities, URL: https://www.rfc-editor.org/rfc/rfc882

11 RFC 883: Domain Names — Implementation and Specification, URL: https://www.rfc-editor.org/rfc/rfc883

12 RFC 1034: Domain Names Concepts and Facilities, URL: https://www.rfc-editor.org/rfc/rfc1034

13 RFC 1035: Domain Names Implementation and Specification, URL: https://www.rfc-editor.org/rfc/rfc1035

example Домен example, содержащий информацию обо всех поддоменах, а также имена серверов, зарегистрированных непосредственно в этом домене, например www.example.com.

web Имя www-сервера и соответствующие ему IP-адреса.

Каждая зона обслуживается двумя или более серверами, отвечающими на запросы клиентов.

В DNS существует три основных типа серверов. Их роль в процессе разрешения имен показана на рис. 18.

1. Авторитетные серверы, которые обслуживают определенные зоны и дают ответы, так сказать, из первых рук. Авторитетные серверы, в свою очередь, делятся на два типа. «Мастер-сервер», его еще называют первичным, непосредственно обслуживает данные зоны. А вторичные серверы зеркалируют эти зоны для улучшения устойчивости и производительности системы. Как правило, первичный сервер обслуживается администратором зоны, а вторичные серверы — по соглашению другими операторами или компаниями, специализирующимися на оказании такого вида услуг. В ответ на запрос авторитетный сервер может либо предоставить запрашиваемую информацию (например, адрес хоста, соответствующего имени обслуживаемого домена), либо выдать отрицательный ответ, если запрашиваемое имя отсутствует в домене. Наконец, сервер может предоставить так называемый реферал (от англ. referral), или перенаправление, указав на серверы, обслуживающие поддомены, содержащиеся в имени.
2. Резолверы, также известные под именем итеративные резолверы. Эти серверы, как правило, обслуживают множество клиентов и выполняют за них основную работу по трансляции имен, а также кешируют полученные ответы для повышения производительности.
3. Резолверы-заглушки, которые выполняют простую функцию — преобразуют запрос приложения в DNS-запрос и передают его серверу, обычно итеративному резолверу, для последующей обработки. При получении ответа они делают необходимые преобразования и передают его обратно приложению. Как правило, резолверы-заглушки реализованы в виде программных библиотек или являются частью операционной системы.

Чтобы проиллюстрировать работу DNS, рассмотрим процесс преобразования имени в соответствующий ему IP-адрес. На рис. 18 приведена схема процесса[14], который происходит, когда вы набираете имя веб-сайта в окошке вашего браузера.

[14] Поэкспериментировать с различными примерами разрешения имени вам поможет утилита dig (http://ru.wikipedia.org/wiki/Dig), предоставляющая пользователю интерфейс командной строки для обращения к системе DNS. Например, процесс на рис. 18 можно увидеть с помощью команды dig + trace www.example.com (подставьте вместо www.example.com какое-либо существующее имя).

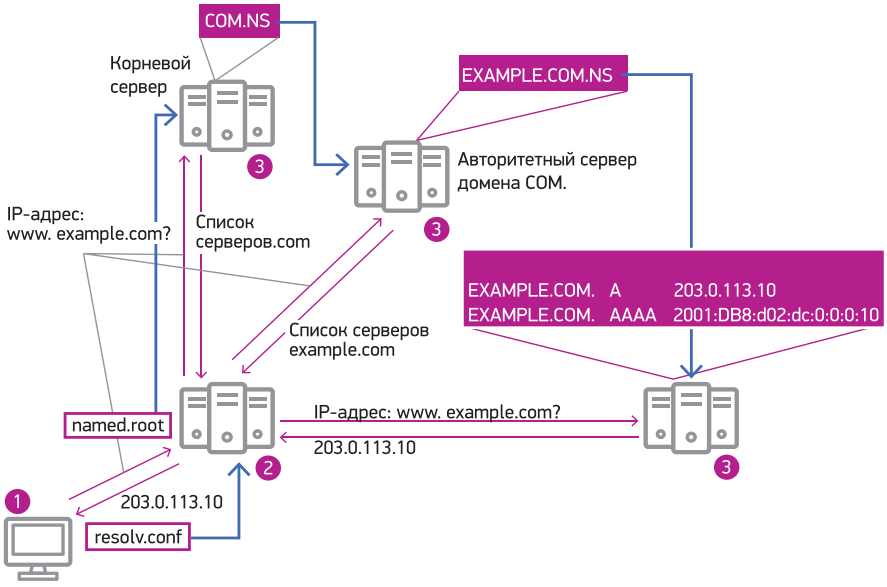

Рис. 18. Процесс трансляции имен в DNS.

Как мы уже говорили, доменные имена — это удобный способ указания ресурса. Для создания соединения TCP/HTTP нужен IP-адрес сервера, поэтому браузер должен обратиться к DNS с вопросом — каков IP-адрес сервера www.example.com (1)?

Процесс разрешения имени может быть достаточно трудоемким — для получения окончательного ответа клиенту необходимо опросить отдельные доменные базы данных, или зоны, соответствующие компонентам полного доменного имени, сужая поиск. Поэтому приложение обычно делает это не самостоятельно, а с использованием итеративного резолвера, который в ответ на полученный запрос выдает конечный результат (2).

Для получения ответа резолверу предстоит опросить все авторитетные серверы, отвечающие за соответствующие части полного доменного имени (3).

Предположим, что основная масса информации от предыдущих запросов в кеше резолвера отсутствует. В этом случае единственное, что знает резолвер в отношении полученного запроса, — адреса так называемых корневых серверов, обслуживающих корневую зону. Эти адреса содержатся в специальном файле, так называемом root hints[15]. О корневой зоне и системе корневых серверов мы

[15] В зависимости от операционной системы и программного обеспечения этот файл может иметь различные названия: named.ca, named.root, root.hints.

поговорим более подробно в разделе «Корневой уровень DNS», сейчас же лишь отметим, что данный файл содержит адреса всех 13 корневых серверов. Мастер-копия этого файла по-прежнему хранится на сайте ftp.internic.net, но резолверы автоматически поддерживают его актуальность, периодически запрашивая корневые же серверы об их текущих адресах — это так называемый первичный, или priming-запрос.

Итак, чтобы начать поиск, резолвер посылает запрос одному из корневых серверов. Эти серверы ничего не знают о существовании домена example.com и тем более адреса www.example.com. Но они сообщат, как можно связаться с серверами, обслуживающими домен следующего уровня — .com.

От одного из серверов .com резолвер узнает адреса серверов домена www.example.com, которые, в свою очередь, и ответят на запрос об IP-адресе сервера www.example.com

Хотя администратор каждой зоны действует достаточно независимо, необходима определенная координация между администраторами дочерней и родительской зон. Например, поскольку родительская зона указывает на серверы, обслуживающие дочернюю, изменения в их составе должны быть своевременно отражены в родительской зоне.

Зоны и записи

Рассмотренный нами принцип работы DNS довольно прост. Попробуем теперь более подробно изучить, что же происходит на уровне протокола.

Начнем с файла простой зоны домена www.example.com.

Таблица 1. Зона домена example.com

Имя	TTL	Класс	Тип записи	Значение
$ORIGIN example.com.				; «точка отсчета», от которой определяются все относительные имена
$TTL 86400				
@	IN	SOA	ns1.example.com.	hostmaster.example.com. (
			2001062501	; серийный номер зоны
			21600	; обновлять каждые 6 ч
			3600	; следующая попытка после 1 ч
			604800	; срок годности 1 неделя
			86400)	; минимальный TTL 1 день
	IN	NS	ns1.example.com.	; серверы имен, обслуживающие
	IN	NS	ns2.example.com.	; домен example.com

Имя		TTL	Класс	Тип записи	Значение
		IN	MX	10 mail1.example.com.	; почтовые серверы, обслуживающи
		IN	MX	20 mail2.example.com.	; домен example.com
ns1		IN	A	203.0.113.1	; IPv4 и IPv6 адреса
ns2		IN	A	203.0.113.2	; серверов имен, обслуживающих
		IN	AAAA	2001: DB8: D02: DC:0:0:0:2	; домен
server1	600	IN	A	203.0.113.10	; значение TTL уменьшено до 10 мин
server2		IN	A	203.0.113.11	; IPv4 и IPv6 адреса
			AAAA	2001: DB8: D02: DC:0:0:0: B	; серверов приложений
ftp		IN	A	203.0.113.12	
		IN	A	203.0.113.13	
mail1		IN	CNAME	server1	; псевдонимы канонического
mail2		IN	CNAME	server2	; имени server1 и server2
www		IN	CNAME	server1	
subdomain		IN	NS	ns1.example.net	; серверы имен, обслуживающие
		IN	NS	ns2.example.net	; поддомен subdomain

Каждая строка этого файла представляет собой так называемую запись ресурса (resource record). Формат записи прост:

Имя	TTL	Класс	Тип записи	Значение

Запись в столбце «Имя» определяет имя в домене зоны, например, www. При этом в нашем примере полное имя ресурса будет www.example.com. Точка в конце имени очень важна и означает полное, в отличие от относительного (относительно определения $ORIGIN) имени.

TTL — время жизни (Time To Live) записи в кеше резолвера в секундах. По прошествии этого времени запись удаляется из кеша. Значение 0 означает, что запись не должна кешироваться вообще, что, безусловно, повышает нагрузку на сервер имен, обслуживающий зону. Ведь каждый запрос клиента на разрешение этого имени будет вызывать обращение резолвера к данному авторитетному серверу.

Следующее поле — «Класс», теоретически позволяющий создавать параллельные деревья DNS. Но поскольку эта особенность практически не используется, мы не будем на ней останавливаться. В большинстве случаев на этом месте вы увидите значение IN — от Internet.

Тип записи (Record Type или RRTYPE) определяет семантику и синтаксис значения записи. Например, значение записи NS — это имя сервера имен, обслуживающего домен. Запись CNAME имеет тот же синтаксис, но ее значение является каноническим именем ресурса. В нашем примере эта запись используется для создания псевдонимов — других имен одного и того же сервера.

В таблице 2 приведены наиболее распространенные типы записей. Часть из них имеет отношение к DNSSEC, расширениям безопасности DNS, о которых мы поговорим чуть позже.

Таблица 2. Наиболее распространенные записи ресурсов DNS

Запись RR	RFC	Описание
A	RFC 1035	IPv4-адрес хоста.
AAAA	RFC 3596	IPv6-адрес хоста.
CNAME	RFC 1035	Canonical Name (каноническое имя). Позволяет определить альтернативные имена хоста (псевдонимы). Результатом ее использования является перенаправление для единственной записи.
DNAME	RFC 6672	Так же, как и CNAME, определяет перенаправление, но на уровне целой ветви DNS.
DNSKEY	RFC 4034	Определяет открытый ключ в DNSSEC.
DS	RFC 4034	Delegated Signer (делегирование подписи). Является указателем на открытый ключ дочерней зоны в DNSSEC.
MX	RFC 1035	Mail Exchanger (почтовый обмен). Определяет приоритет и имя почтового сервера, обслуживающего электронную почту для данной зоны.
NAPTR	RFC 3403	Naming Authority Pointer (указатель авторитетных имен), название не имеет ничего общего с функциональностью записи, на самом деле она используется для определения правил так называемой системы обнаружения динамического делегирования (Dynamic Delegation Discovery System, DDDS), например, при использовании протоколов VoIP или ENUM для передачи голоса по IP.
NS	RFC 1035	Name Server (сервер имен). Определяет имя авторитетного сервера имен для зоны.
NSEC	RFC 4034	Next Secure (следующая защищенная запись). Используется для подтверждения отсутствия записи в DNSSEC.
NSEC3	RFC5155	Так же, как и NSEC, используется для подтверждения отсутствия записи имени в DNSSEC, но за счет использования хешей вместо имен последующих записей предотвращает возможность получения содержимого зоны — так называемой прогулки по зоне (zone walking).
PTR	RFC 1035	Определяет имя, соответствующее IP-адресу (IPv4 или IPv6); используется в «обратном» DNS.
RRSIG	RFC 4034	Signed RRset (подписанная запись ресурса) в DNSSEC.
SOA	RFC 1035	Start of Authority (указание на авторитетность информации). Определяет имя зоны, контактный адрес электронной почты и различные параметры зоны по умолчанию: частоту обновления, «срок годности» зоны и отдельных записей.
SPF	RFC 4408	Sender Policy Framework (система политики отправителя). Определяет почтовые серверы, с которых может быть отправлена почта домена, используется для борьбы против спуфинга — маскировки и фальсификации почтового адреса-источника при посылке спама.
SRV	RFC 2872	Определяет дополнительные услуги, связанные с доменом, например, ldap, http, sip; обычно используется совместно с записью NAPTR, поддерживая обнаружение серверов этих дополнительных услуг.
TXT	RFC 1035	Текстовая информация, ассоциированная с именем, все чаще используется для расширения функциональности DNS, требуемой новыми протоколами и системами, такими как SPF или DKIM, поскольку создание и внедрение дополнительных типов записей DNS становится все более затруднительным.

Линии отреза и делегирование в DNS

Прежде чем говорить о запросах и ответах на них, кратко остановимся на том, как происходит делегирование в DNS. Как мы уже обсуждали, различные домены могут обслуживаться различными администраторами. В то же время структура DNS иерархическая, и существуют понятия родительской и дочерней зоны. Администратор родительской зоны может делегировать часть пространства имен другому администратору. Место в зоне, где происходит делегирование, называют «линией отреза» (zone cut).

В DNS эта линия может проходить только между компонентами имени. Другими словами, можно делегировать поддомен «subdomain», но нельзя делегировать все имена третьего уровня, начинающиеся с буквы s. Точнее говоря, это возможно, но для каждого имени придется задавать отдельную делегацию и, соответственно, отдельную дочернюю зону.

На первый взгляд, данная проблема носит, скорее, теоретический характер. Однако она требует практического решения в случае конфигурации обратной зоны DNS, необходимой для обслуживания так называемых обратных запросов — трансляции IP-адреса в доменное имя. Чтобы лучше понять, как происходит делегирование в DNS, давайте посмотрим на «обратный» DNS более подробно. Но сначала будет уместно сказать несколько слов о том, что такое обратная зона и обратные запросы.

Термин «обратный» используется для контраста с обычными, прямыми запросами DNS на трансляцию имени в адрес IP. При обратном запросе происходит поиск соответствующего имени для указанного IP-адреса. Зачем это может потребоваться? Например, вывод утилиты traceroute будет более «читабельным», если вместо адресов будут показаны соответствующие имена узлов, через которые проходит трафик. Обратный DNS также используется как форма аутентификации клиента — если запрос приходит от хоста с именем, это указывает на существование административного домена и его обслуживание. Данный подход используется, например, в фильтрации запросов от спамеров с компьютеров-зомби.

Однако для работы обратных запросов недостаточно конфигурации прямой зоны: единственным ключом поиска в DNS является имя, а поиск по значению (например, по IP-адресу) невозможен. Для решения этой проблемы была создана отдельная ветвь — «обратный» DNS — и специальный тип записи PTR. Ветвь эта берет свое начало в домене in-addr.arpa.

Запись PTR выглядит следующим образом:

```
$ORIGIN 113.0.203.in-addr.arpa.
...
10      IN      PTR     server1.example.com.    ; полное доменное имя сервера
```

Отправляя запрос на трансляцию имени 203.0.113.10 (не правда ли, оно похоже на доменное имя?), резолвер осуществит поиск в «обратном» DNS. Правда, сначала он преобразует адрес в «обратное» доменное имя: 10.113.0.203.in-addr.arpa. Вы, наверное, догадались, каким образом: компоненты адреса IPv4 перечислены в обратном порядке, и к полученной строке приставлен домен in-addr.arpa. Таким образом, поиск будет осуществлен сначала в корневой зоне, затем — в arpa., in-addr.arpa. и, наконец, в 113.0.203.in-addr.arpa., где и найдется запись с именем 10 и значением server1.example.com.

Все это прекрасно работало до внедрения CIDR — системы маршрутизации и соответствующей ей системы распределения адресного пространства, о которой мы говорили в предыдущей главе. Делегирование проводилось по границе октетов (каждые 8 бит адреса), что соответствовало его текстовому десятичному представлению. С появлением CIDR ситуация изменилась: стало возможным выделить клиенту сеть, скажем, размером /25. Предполагая, что мы также передаем административный контроль за обратным DNS, как же будет выглядеть обратная зона для этой сети?

Здесь и возникает проблема делегирования, которое возможно только по границе компонентов имени. То есть для 203.0.113.10 можно, например, делегировать последний компонент — 10, но это будет соответствовать сети /24!

Решение было предложено в 1998 году в RFC 2317[16], оно основано на применении псевдонимов и записей CNAME. Трюк заключается в создании промежуточного псевдодомена, например, «0–127», и его делегировании администратору сети /25. Например:

```
$ORIGIN 113.0.203.in-addr.arpa.
...
0–127   IN    NS      ns1.domainA.example.org.   ; полное доменное имя сервера
        IN    NS      ns2.example.com.           ; имен, обслуживающего зону
...
1       IN    CNAME   1.0–127.113.0.203.in-addr.arpa.
2       IN    CNAME   2.0–127.113.0.203.in-addr.arpa.
3       IN    CNAME   3.0–127.113.0.203.in-addr.arpa.
...
```

Соответственно, делегированная обратная зона для сети 203.0.113.0/25 будет выглядеть следующим образом:

[16] RFC 2317: Classless IN-ADDR.ARPA delegation, URL: https://www.rfc-editor.org/rfc/rfc2317

```
$ORIGIN 0–127.113.0.203.in-addr.arpa.
...
@       IN    NS      ns1.domainA.example.org.   ; полное доменное имя сервера
I       N     NS      ns2.example.com.
...
        IN    PTR     host1.domainA.example.org.
        IN    PTR     host2.domainA.example.org.
        IN    PTR     host3.domainA.example.org.
...
```

Возможно, вы задаете себе вопрос: как «обратный» DNS работает в случае с IPv6? Приведем пример записи PTR для такого случая. IPv6 использует собственную ветвь в ipv6.arpa, а декомпозиция адреса производится на каждые 4 бита, а не 8, как в IPv4.

```
$ORIGIN c.d.0.0.2.0.d.0.8.b.d.0.1.0.0.2.ip6.arpa.
...
b.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0   IN   PTR   server2.example.com.   ; полное доменное
                                                                          имя сервера
```

Запросы и ответы в DNS

Работа DNS происходит по схеме «запрос–ответ». Поскольку размеры запросов и ответов достаточно невелики, в основном используется транспортный протокол UDP, который, в отличие от TCP, не требует создания соединения и обеспечивает минимальные накладные расходы. Правда, ситуация постепенно меняется: размер ответов становится все больше, а с внедрением DNSSEC он может легко превысить несколько килобайт. Но об этом — позже.

И запрос, и ответ имеют одинаковую структуру, состоящую из пяти разделов: заголовок, запрос, ответ, авторитет и дополнительная информация. Эта структура представлена на рис. 19.

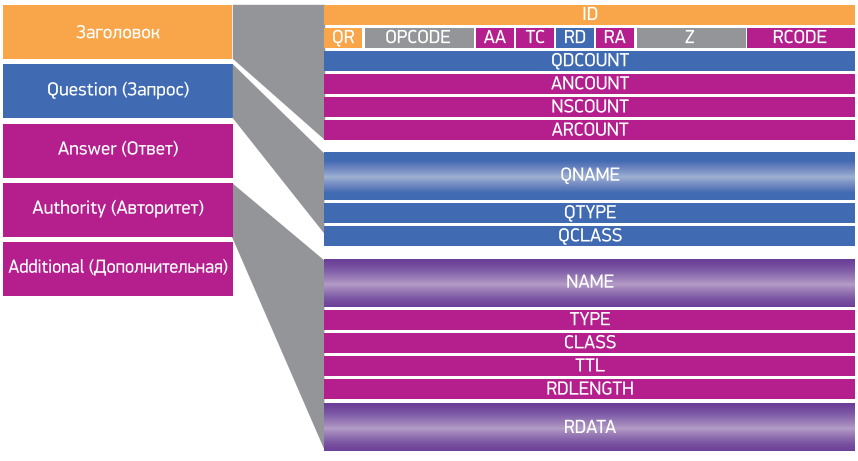

Рис. 19. Структура сообщения DNS.

Запрос использует только заголовок и первый раздел сообщения, который состоит из трех основных элементов: QNAME, QTYPE и QCLASS.

QNAME — доменное имя, именно оно в DNS всегда является поисковым ключом. Например, www.example.com.

QTYPE — тип требуемой записи, связанной с именем, указанным в QNAME. Существует также мета-тип — ANY, указывающий на запрос всех записей, связанных с именем QNAME.

QCLASS — класс: как уже говорилось, в основном это класс IN. Клиент также может указать дополнительную информацию. Например, флаг RD (Recursion Desired, «желательна рекурсия») — инструкция для запрашиваемого сервера имен, по которой следует произвести рекурсивный запрос, то есть провести полный процесс разрешения имени, показанный на рис. 18, до получения окончательного ответа. Данный флаг используется при обращении клиента (или резолвера-заглушки) к итеративному резолверу.

В зависимости от типа ответа в сообщении DNS, помимо заголовка, используются разделы ответа, авторитета и дополнительная информация. Все они имеют один и тот же формат, показанный на рис. 19.

Вы заметите, что этот формат почти в точности соответствует формату самой записи: и NAME, TYPE, CLASS, TTL и RDATA соответствуют имени, типу записи, классу, TTL и значению. RDLENGTH указывает на размер поля «Значение» записи.

Раздел ответа содержит записи, соответствующие запрашиваемому имени, если таковые найдены. В противном случае этот раздел будет пустым. Если сервер уверен, что имя не существует, то в заголовке будет установлен флаг RCODE — NXDOMAIN, указывающий, что имя не найдено. Эта ситуация отлична от случая, когда имя найдено, но не найдены запрашиваемые записи, или если имя выходит за переделы «линии отреза» и может присутствовать в дочерней зоне. В этих случаях флаг ошибки установлен не будет. Если ответ получен от авторитетного сервера для зоны, в которой происходит поиск имени, то в заголовке сообщения будет установлен флаг AA (Authoritative Answer, авторитетный ответ). Ответ, полученный из кеша, например, от итеративного резолвера, не содержит этого флага. В этом случае поле TTL будет указывать не на общий срок годности, а на оставшееся время жизни записи в кеше.

Раздел	NAME	TTL		RTYPE	RDATA
Ответ	server1.example.com.	1800	IN	A	203.0.113.11

Записи авторитетного раздела указывают на авторитетные серверы, где следует продолжить поиск для получения окончательного ответа. Такой ответ еще называют referral (реферал, или ссылка). Наиболее типичным является получение реферала от серверов доменов верхних уровней. В нашем примере (рис. 18) — это ответ от корневого сервера и от сервера домена com.

В дополнительный раздел сервер помещает информацию, которая клиенту, вероятнее всего, понадобится в будущем. Например, сервер домена .com при передаче реферала, указывающего на серверы ns1.example.com и ns2.example.com в авторитетном разделе, в дополнительном разделе укажет их IP-адреса. На запрос

Раздел	QNAME	QTYPE	QCLASS
Запрос	www.example.com.	A	IN

один из серверов домена .com (например, c.gtld-servers.net.) ответит:

Раздел	NAME	TTL		RTYPE	RDATA
Авторитетный	example.com.	86400	IN	NS	ns1.example.com.
	example.com.	86400	IN	NS	ns2.example.com.
Доп	ns1.example.com.	86400	IN	A	203.0.113.1
	ns2.example.com.	86400	IN	A	203.0.113.2

Интернационализация DNS

Система и соответствующие протоколы DNS были созданы для замены файла hosts.txt, который содержал перечень всех хостов Сети, существовавших на тот момент. Система репликации этого файла на всех компьютерах Сети не масштабировалась, а число хостов неуклонно росло. Поддержка уникальности имен, разумного времени синхронизации и обновления этих данных в глобальном масштабе — вот основные требования, предъявлявшиеся тогда к DNS. Учитывая, что все развитие Сети в то время происходило почти исключительно в англоязычной среде, требования многоязыковой поддержки не было.

Однако по мере роста и расширения Интернета ситуация начинала меняться. Требование именовать ресурсы исключительно символами ASCII казалось все более ограничительным и не соответствующим уровню глобализации Интернета.

Как сказано в одном из документов «Консорциума Юникода» — некоммерческой организации, отвечающей за разработку и сопровождение стандарта кодирования символов национальных алфавитов Юникод[17]:

[17] https://ru.wikipedia.org/wiki/Юникод

«Изначально доменные имена были ограничены набором символов ASCII. Это являлось существенным неудобством для людей, использующих другие символы. Представьте, например, что система доменных имен была бы изобретена греками и, соответственно, в URL приходилось бы использовать только греческие символы. Вместо apple.com пользователи вынужденно набирали бы что-нибудь типа αππλε.κομ. Англоговорящие пользователи должны были бы не только знать греческий алфавит, но и уметь выбирать греческие символы в соответствии с желаемым английским именем. Пришлось бы гадать, что означает то или иное имя, ведь написания не совпадают[18]».

Эта довольно абсурдная ситуация существовала до относительно недавнего времени и была актуальна для неанглоязычных пользователей Интернета. И хотя попытки интернационализировать доменные имена и предоставить возможность задавать имена на родном языке начались еще в конце 90-х годов прошлого столетия, реальная работа по стандартизации глобально применимых решений в IETF началась только в 2000 году с созданием рабочей группы по IDN (Internationalized Domain Names, интернационализированные доменные имена).

IDNA 2003

В ходе разработки стандартов рабочей группой были приняты следующие основополагающие решения:

1. Использовать Юникод как наиболее полный набор алфавитов и символов для ввода и отображения интернационализированных доменных имен. Юникод представляет собой систему кодирования символов национальных письменных языков и поддерживается «Консорциумом Юникода». Стандарт был совместно разработан с Международной организацией по стандартизации ISO, которая также поддерживает его под именем ISO/IEC 10646. Хотя оба стандарта синхронизируются, между ними есть небольшие отличия в плане требований использования и метаданных. Стандарт Юникод постоянно обновляется, в него добавляются новые языки и символы. На момент написания этой книги последней версией стандарта является Unicode 9.0[19].
2. Минимизировать изменения в архитектуре DNS, программном обеспечении элементов DNS — серверов и резолверов. Интересно отметить, что сам протокол DNS передает имена и другие поля в бинарном виде, и практически единственное ограничение, которое накладывает спецификация на имя, — это его длина. Длина имени (каждого компонента полного доменного имени) не должна превышать 63 байт, а общая длина с учетом разделителей — «.» — не должна быть больше 255 байт[20]. Широко распространенные ограничения на символы, из которых составляется доменное имя, так назы-

[18] «Unicode IDNA Compatibility Processing», http://unicode.org/reports/tr46
[19] http://www.unicode.org/versions/Unicode9.0.0
[20] RFC 2181: Clarifications to the DNS Specification, URL: https://www.rfc-editor.org/rfc/rfc2181

ваемые LDH (Letter, Digit, Hyphen — буква, цифра, дефис в формате ASCII), диктуются не самим протоколом, а приложениями, которые используют эти имена. LDH, таким образом, является «наименьшим общим знаменателем» всех ограничений. Для достижения максимальной совместимости было решено взять правило LDH в качестве требования к интернационализированному имени на уровне протокола DNS.

Таким образом, вся поддержка интернационализированных доменных имен происходит вне самой системы DNS, исключительно на уровне приложений и пользовательского интерфейса. А задача сводится к трансляции символов Юникода в формат ASCII с ограничением LDH.

Для трансляции символов Юникода в набор допустимых символов DNS был выбран алгоритм под названием Пьюникод (Punycode). Этот алгоритм определен в стандарте IETF RFC 3492[21]. Имена в Пьюникоде выглядят немного странно: например, слово «испытание» будет представлено как xn-80akhbyknj4f. Но зато они полностью соответствуют стандарту LDH. Для указания на то, что имя является ASCII-представлением (или ACE — ASCIIcompatible encoding) интернационализированного имени, используется специальный префикс xn-. Этот же алгоритм позволяет декодировать ACE-имя и получить обратно исходное слово в формате Юникод. Казалось бы, выход найден. Однако требовалось решить еще одну проблему. Дело в том, что DNS по определению осуществляет поиск до точного совпадения. Это означает, что имя в запросе (QNAME) должно полностью соответствовать имени ресурса. За исключением того, что сравнение производится без различия строчных и прописных букв.

В письменном языке все по-другому, особенно если символы кодируются в Юникоде. Отдельные символы могут выглядеть почти идентично, но иметь различные коды Юникода. Как отмечают авторы документа «Обзор и рекомендации относительно интернационализованных доменных имен»[22], символы ø (U+00F8 — LATIN SMALL LETTER O WITH STROKE) и ö (U+00F6 — LATIN SMALL LETTER O WITH DIAERESIS) считаются, например, эквивалентными в шведском языке. В то же время коды этих символов различны, соответственно, будут различны и имена с точки зрения DNS. Другой пример дополнительной сложности — некоторые символы Юникода невидимы или почти невидимы: с виду идентичные имена будут расценены системой DNS как разные. Наконец, нечувствительность DNS к регистру (другими словами, для DNS не имеет значения, строчными или прописными буквами написано имя) следует также реализовать на уровне приложений, поскольку результат трансляции Пьюникодом строчных и прописных букв будет различен. И протокол DNS, которому ничего не известно о Юникоде или Пьюникоде, будет рассматривать их как два разных имени.

---

[21] RFC 3492: Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA), URL: https://www.rfc-editor.org/rfc/rfc3492

[22] RFC 4690: Review and Recommendations for Internationalized Domain Names (IDNs), URL: https://www.rfc-editor.org/rfc/rfc4690

Для преодоления данных сложностей рабочей группой IDN был разработан процесс приведения в соответствие имени, набранного пользователем, включая отображение «совместимых» символов[23], приведение всех символов к строчному регистру, а также исключение некоторых символов. Этот процесс получил название Nameprep и был стандартизован в RFC 3491[24].

Группа стандартов, определяющих процесс обработки интернационализированных имен в приложениях, получила общее название IDNA 2003 — по году, когда были опубликованы составляющие ее RFC: IDNA RFC 3490[25], Nameprep RFC 3491, Punycode RFC 3492, Stringprep RFC 3454[26]. Этот процесс показан на левой стороне рис. 20.

Однако такой подход имел несколько существенных недостатков, которые становились все более очевидными по мере появления интернационализированных доменов и внедрения IDNA 2003. Как сказано в RFC 5895[27], «изначальная версия IDNA объединила и обработку на уровне пользовательского интерфейса, и собственно протокол. Она принимала любые символы, набранные пользователем, в кодировке, поддерживаемой его приложением, обеспечивала преобразование в коды Юникода, а затем без учета контекста, локальных настроек и какого-либо знания о намерениях пользователя отображала их в определенный набор других символов. И в конечном счете IDNA 2003 перекодировала эти символы Пьюникодом. Игнорирование контекста, языковых и пользовательских предпочтений в протоколе IDNA значительно облегчила жизнь разработчикам приложений. Но для потребителей и производителей доменных имен возросла вероятность некорректной трансформации исходного запроса, что явным образом нарушало так называемый принцип наименьшего сюрприза».

Пожалуй, еще более существенным недостатком IDNA 2003 явилось то, что стандарты были жестко привязаны к версии Юникода 3.2. Соответственно, при появлении новых версий (а версия 4 появилась уже в 2003 году) требовалось обновление стандартов — процесс достаточно трудоемкий.

Некоторые домены верхнего уровня (.jp, .info и несколько других) уже позволяли регистрацию интернационализированных имен, но указанные недостатки привели к тому, что в 2008 году IETF начал работу над новой серией стандартов IDNA (IDNA-bis), которая впоследствии получила имя IDNA 2008.

---

23 См. формы нормализации Юникода: http://unicode.org/reports/tr15

24 RFC 3491: Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN), URL: https://www.rfc-editor.org/rfc/rfc3491

25 RFC 3490: Internationalizing Domain Names in Applications (IDNA), URL: https://www.rfc-editor.org/rfc/rfc3490

26 RFC 3454: Preparation of Internationalized Strings («stringprep»), URL: https://www.rfc-editor.org/rfc/rfc3454

27 RFC 5895: Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008, URL: https://www.rfc-editor.org/rfc/rfc5895

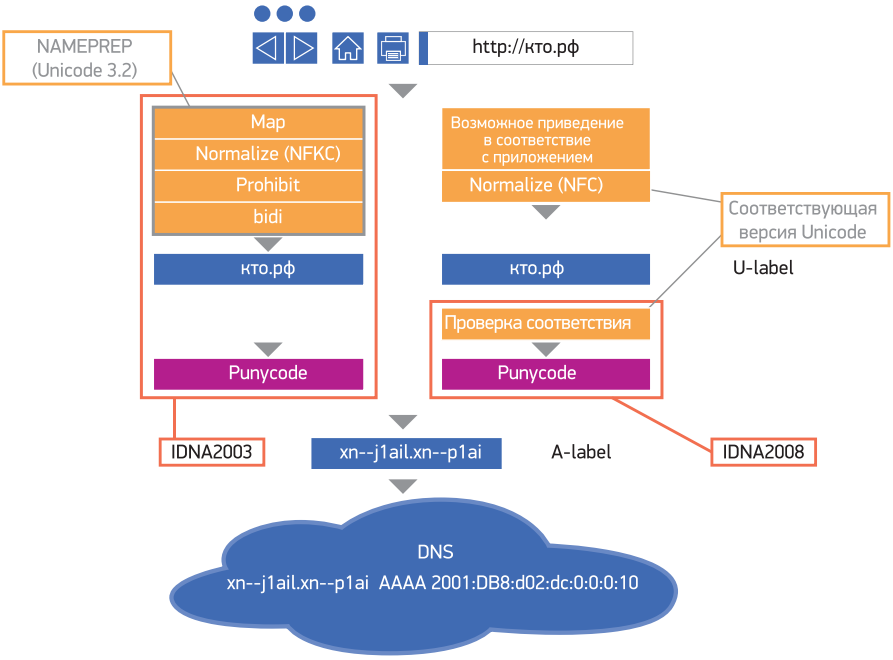

Рис. 20. Процесс обработки интернационализированных доменных имен в стандартах IDNA 2003 и IDNA 2008.

IDNA 2008

Разработчики новых стандартов выбрали несколько другой подход. Вместо того чтобы допускать практически любой символ Юникода, а затем пытаться путем отображений получить строку, готовую к трансляции в ACE, и которая, в лучшем случае, является догадкой о намерениях пользователя, в IDNA 2008 действует так называемый принцип включения. Согласно этому принципу код является недопустимым, если только он не соответствует стабильным правилам, определенным стандартам, или в редких случаях явным исключениям. К стабильным правилам относится, например, определение принадлежности символа к классу «буква или цифра». Важным здесь является то, что такие правила не зависят от используемой версии Юникода.

В результате стандарт IDNA 2008 запретил более 8000 символов, допустимых в стандарте IDNA 2003. Это ограничение репертуара символов Юникода повысило стабильность трансформации запросов, за счет чего признано обоснованным. Многие ассоциируют имена в DNS со словами, но на деле последние являются не более чем мнемониками. Соответственно, задачей IDNA является не обеспечение возможности «написать новеллу на клингонском

(или любом другом) языке[28], используя доменные имена, а поддержка создания полезных естественных мнемоник для очень широкого диапазона письменных языков»[29].

Предполагается, что приложение само, в рамках местного контекста и предпочтений пользователя, обеспечит приведение строки запроса к виду, допустимому в IDNA 2008. В качестве примеров можно перечислить кодирование строки в Юникод, если при вводе использовалась другая кодировка, приведение строки к строчным символам (IDNA 2008 включает только строчные символы), а также нормализация Юникода NFC[30].

Существенные различия между стандартами IDNA 2003 и IDNA 2008 показаны на рис. 20.

Заметим, что с использованием IDNA связан и ряд проблем. Например, IDNA открывает более широкие возможности для спуфинга (spoofing) имен, когда имя сервера внешне очень похоже на другое имя, но на самом деле использует другие символы, так называемые гомографы. Действительно, как отличишь paypal.com от paypal.com, в котором вторая буква «а» набрана кириллицей? По сравнению с обычными именами, где 1 похоже на l, а o на O, IDNA содержит гораздо больше гомографов. Решение этой проблемы требует строгого контроля со стороны регистраторов доменов, ограничения числа поддерживаемых языков в рамках домена и запрета смешивания различных языков в доменном имени.

Вопросы безопасности DNS

Итак, мы уже хорошо знаем, что система DNS является иерархической и распределенной. Оператор каждой зоны может самостоятельно определить необходимые ресурсы для обеспечения стабильности и производительности. Это делает систему в целом устойчивой и масштабируемой.

Однако специалисты в области компьютерной безопасности определяют широкий спектр угроз — потенциальных атак на систему DNS, использующих различные уязвимые места системы. Некоторые векторы атак потенциально могут нарушить функционирование глобальной DNS и, как следствие, глобального Интернета. Другие угрозы направлены больше на отдельные организации и группы пользователей.

Рассмотрим их подробнее.

---

28 https://ru.wikipedia.org/wiki/Клингонский_язык

29 RFC 5894: Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale, URL: https://www.rfc-editor.org/rfc/rfc5894

30 Каноническая форма, http://unicode.org/reports/tr15

Угрозы и уязвимые места DNS

Угрозы DNS и связанные с ними атаки можно разделить на две основные категории: **атаки отказа в обслуживании** и атаки модификации данных DNS.

Цель атаки отказа в обслуживании (DoS, Denial of Service) — сделать недоступным разрешение имен для отдельных доменов. В своем типичном варианте атака генерирует громадный объем трафика, направленный на атакуемый ресурс. Это приводит к истощению ресурсов серверов домена или всей сетевой инфраструктуры, обеспечивающей его работу.

Отказ в обслуживании домена может привести к отказу обслуживания и всех дочерних доменов, связанных с ним. Также длительная недоступность первичного сервера может привести к истечению срока действия зоны на вторичных серверах и как следствие — к исчезновению зоны (и всей иерархической инфраструктуры ниже) для пользователей.

Атаки DoS, чаше всего имеющие распределенный характер, когда источники атаки расположены в различных точках Интернета (Distributed DoS, DDoS), являются атаками общего типа и могут быть направлены против любого ресурса Интернета. Интересной особенностью здесь является то, что DNS может являться как жертвой, так и средством проведения распространенного типа атаки — атаки усиления. Об этом мы поговорим чуть позже, в разделе «Противодействие атакам усиления».

Другой тип угроз — **атаки, связанные с подменой и модификацией данных DNS**. В то же время основной недостаток базового протокола DNS — слабая система защиты данных. В процессе передачи данных от сервера к клиенту они могут быть модифицированы. За счет изменения данных DNS можно создавать ложные почтовые серверы, перехватывать и перлюстрировать почтовые сообщения пользователей этого сервера. Другим примером является создание злоумышленниками веб-сайтов, имитирующих услуги электронных магазинов, банков, государственных учреждений. Последствия могут быть долгосрочными и значительными как для отдельной группы пользователей, так и для отдельного сегмента сети Интернет.

Эффективным средством борьбы с эти типом атак выступает применение расширений безопасности DNS, разработанных в рамках IETF[31] и получивших название DNSSEC.

Чтобы лучше понять значимость этой технологии для безопасности DNS, рассмотрим, насколько уязвима система в целом. Различные уязвимые места системы показаны на рис. 21, а методы защиты — на рис. 22.

[31] Internet Engineering Task Force, http://www.ietf.org

Первый вопрос — насколько защищен собственно процесс подготовки данных, редактирования и создания зоны DNS? Ошибочные данные, умышленно или случайно оказавшиеся в зоне (например, неправильные адреса веб-сайта или почтовых серверов), будут переданы пользователю в ответ на его запрос независимо от использования DNSSEC. В данном случае значение имеет уровень внутренней безопасности и защищенности процесса.

Рис. 21. Уязвимые места DNS.

Рис. 22. Методы защиты DNS.

Данные также могут быть модифицированы при передаче от мастер-сервера ко вторичным DNS-серверам, обслуживающим зону. Сегодня для проверки целостности передачи данных используется протокол TSIG (Transaction SIGnature).

Ответы от DNS-серверов также могут быть модифицированы в процессе передачи в Интернете. Но, пожалуй, самое опасное — если «неправильные» данные попадут в кеш итеративных резолверов: это называется «отравление» кеша. Дело в том, что продолжительность жизни данных в кеше, определяемая параметром TTL записи, может быть достаточно значительной. А значит, в течение всего этого времени при «отравлении» кеша пользователи будут получать подложные ответы. Более того, внедрение подложных данных в кеш резолвера не представляет особого труда, как было продемонстрировано Дэном Каминским (Dan Kaminsky) в 2008 году[32]. С тех пор в программное обеспечение большинства стандартных резолверов были введены изменения, усложняющие проведение атаки, но только DNSSEC является единственным эффективным методом защиты, поскольку позволяет криптографически проверить аутентичность данных перед загрузкой их в кеш.

Еще одна возможность публикации ложных данных в DNS — создание лжесерверов DNS. Поскольку адресация и доступ к серверам происходит по их IP-адресу, атака на систему маршрутизации, а именно «захват» соответствующего адресного префикса может привести к перенаправлению части запросов DNS к DNS-серверу злоумышленника. Соответственно, злоумышленник получает возможность публикации ложных данных с далеко идущими последствиями. Такая угроза особенно значительна при масштабном внедрении технологии anycast (аникаст), когда несколько серверов, расположенных в различных частях Интернета, используют один и тот же IP-адрес. В этом случае отличить достоверный DNS-сервер от лжесервера становится сложнее.

Наконец, ответы резолвера на запросы клиентов могут быть также модифицированы. DNSSEC здесь не поможет, если пользователь не производит проверку подлинности ответов самостоятельно, а полагается на резолвер. Правда, канал между резолвером и пользователем, как правило, находится под административным контролем сервис-провайдера или администратора корпоративной сети и зачастую имеет высокую степень защиты, например с помощью VPN.

Повышение устойчивости системы и противодействие атакам усиления

Учитывая критичность системы DNS для нормальной работы Интернета, ее устойчивость имеет большое значение. Различные требования (см., например, Технические требования к авторитетным серверам имен[33] или

[32] https://www.cnet.com/news/privacy/researcher-offers-insight-into-dns-flaw/

[33] Technical requirements for authoritative name servers, https://www.iana.org/help/nameserver-requirements

RFC 1912[34]) устанавливают, что любая зона обслуживается как минимум двумя серверами, которые находятся в разных сетях. Тем самым гарантируется работоспособность в случае отказа одного из компонентов. Также рекомендуется использовать различное программное обеспечение для минимизации ситуаций, когда ошибка или использованная уязвимость ПО приводит к выходу из строя всей системы. Однако отказ или недоступность одного из серверов (например, вследствие потери связности или поломки) являются только одной из проблем. Другой фактор, который может повлиять на устойчивость и производительность системы, — атаки «отказа в обслуживании» DoS. Чтобы эффективно им противостоять, необходимо иметь как можно более распределенную систему серверов.

Также при оценке производительности и устойчивости системы нужно принимать во внимание такие факторы, как устойчивость к перегрузкам и время реакции — промежуток времени, необходимый для получения ответа клиентами, учитывая их географическую распределенность.

Мы коснулись атак DDoS, когда обсуждали угрозы и уязвимые места DNS. Тогда же было отмечено, что в плане атак усиления, которые являются специальным типом атак отказа в обслуживании DDoS, DNS играет двоякую роль.

Во-первых, атаки усиления зачастую выбирают DNS в качестве цели. Атака такого рода может привести к отказу обслуживания не только атакуемого домена, но и всех дочерних доменов, связанных с ним. Это, в свою очередь, сделает практически недоступными информационные ресурсы (например, веб и электронную почту), связанные с соответствующими именами. Во-вторых, DNS сам является ключевым элементом так называемых рефлекторных атак, играя роль усилителя.

Кратко остановимся на самих атаках.

Рефлекторные атаки с усилением

Суть рефлекторной атаки достаточно проста и базируется на трех основных ингредиентах:

1. **Использование возможности «спуфинга»** — подмены IP-адреса отправителя на адрес «жертвы» с протоколами UDP или ICMP, обеспечивающими передачу дейтаграмм без создания соединения. Такие широко распространенные услуги Интернета, как SNMP и DNS, используют именно эти протоколы передачи. Ключевым фактором здесь является отсутствие необходимости «рукопожатия», как, например, в случае с TCP, для начала передачи данных.
2. **Рефлекторы и усилители.** Поскольку режимом работы многих услуг, основанных на протоколе UDP, является «запрос-ответ», при подмене адреса отправителя на адрес «жертвы» ответ на запрос будет доставлен именно

[34] RFC 1912: Common DNS Operational and Configuration Errors, URL: https://www.rfc-editor.org/rfc/rfc1912

туда. Представьте, что такого типа запросы посланы с различных точек Интернета. Все ответы на эти запросы будут направлены на один адрес, обеспечивая значительную концентрацию трафика в направлении «жертвы». «Хороший» рефлектор также является усилителем, когда размер ответа во много раз превышает размер запроса (см. таблицу 2). Это позволяет создать асимметрию, когда относительно незначительный трафик запросов превращается в мощный ответный поток.

3. **Ботнеты.** Для эффективных атак такого рода необходима хорошо распределенная сеть источников. Инфицированные компьютеры, объединенные в ботнеты, являются для этого прекрасной стартовой площадкой.

Смешав эти ингредиенты, мы получим рефлекторную атаку с усилением. Работает она следующим образом.

Таблица 3. Типичные усилители

Протокол	Протокол/Порт	Наблюдаемый коэффициент усиления	Ориентировочное число усилителей в Интернете
DNS	UDP/53	100	7–15 млн
SNMPv2	UDP/161	12	5 млн
NTP	UDP/123	4600	1,5 млн
CHARGEN	UDP/19	360	90 тыс.
SSDP	UDP/1900	80	3,7 млн

Источник: C. Rossow, «Amplification Hell: Revisiting Network Protocols for DDoS Abuse» (https://www.christian-rossow.de/publications/amplification-ndss2014.pdf)

- Выбирается один или несколько усилителей-рефлекторов. В качестве таковых могут служить DNS-серверы и «открытые» резолверы (о них мы поговорим чуть позже), готовые предоставить ответ на запрос со значительным коэффициентом усиления. Типичным является усиление трафика в 30–60 раз.
- Компьютеры ботнета получают инструкцию начать атаку на жертву. По команде они посылают заданные запросы выбранным усилителям-рефлекторам, подменяя адрес отправителя на IP-адрес жертвы.
- Серверы-рефлекторы отвечают на невинные с виду запросы, которые реально поступают от различных клиентов, в то же время обрушивая усиленный трафик на жертву. Поскольку обычно используется большое количество рефлекторов, расположенных в различных точках Интернета, объем трафика увеличивается по мере приближения к жертве.
- Объем сгенерированного трафика превышает пропускную способность каналов или максимальную производительность атакуемого сервера, тем са-

мым вызывая отказ в обслуживании или невозможность предоставления услуг. Услуга, подвергшаяся атаке, на какое-то время перестает быть доступной в Интернете. Этот процесс схематически представлен на рис. 23.

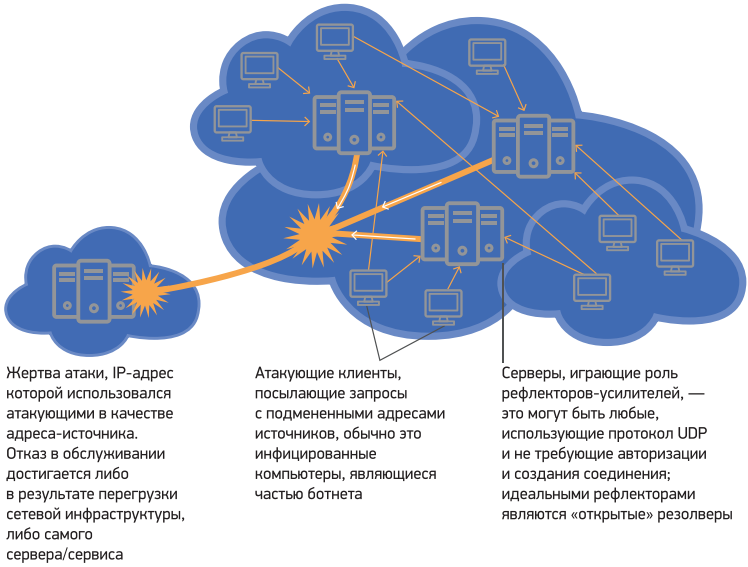

Рис. 23. Схема рефлекторной атаки с усилением.

Поскольку атакующий генерирует незначительный трафик с географически распределенного ботнета, этот трафик трудно идентифицировать и заметить что-либо подозрительное. В подавляющем большинстве случаев владельцы компьютеров ботнета сами не подозревают, что они являются источником атаки. А использование спуфинга адреса отправителя делает задачу определения источника практически невозможной.

С конца 1990-х годов DNS активно используется в качестве рефлектора и усилителя. Действительно, DNS является для этого идеальной услугой:

- DNS использует UDP, не требующий создания соединения и, таким образом, допускающий спуфинг;
- DNS является жизненно необходимой услугой, поэтому фильтрация входящего DNS-трафика практически не применяется;
- DNS является хорошим усилителем трафика, поскольку всегда можно найти запрос, размер ответа на который намного превышает размер самого запроса.

Это последнее свойство было известно с самого начала, однако ограничение максимального размера ответа 512 байт позволяло достичь лишь восьми–девятикратного усиления. Ситуация существенно изменилась с внедрением расширений EDNS0, позволяющих использовать дейтаграммы размером более

512 байт, и DNSSEC, включающего дополнительные данные. Если раньше для создания максимально возможного ответа использовались синтетические записи (например, специально созданная запись txt для определенного домена), то теперь стало возможно генерировать ответы большого размера для запросов, не вызывающих подозрения и использующих вполне безобидные домены.

Например, запрос всех существующих записей для имени isc.org с поддержкой DNSSEC

```
$ dig + dnssec isc.org. any
```

способен сгенерировать ответ размером около 3 килобайт, обеспечивая тем самым 50-кратное усиление (рис. 24).

Не будем забывать, что DNS — распределенная сеть с десятками, если не с сотнями миллионов серверов, обеспечивающих обработку запросов. Меньшая часть — это авторитетные серверы, отвечающие за определенные домены и связанные с ними записи, а подавляющее большинство составляют кеш-резолверы, обслуживающие весь рекурсивный процесс разрешения имен для клиента и предоставляющие последнему окончательный ответ.

Операционная практика предписывает авторитетным серверам отвечать только на запросы по доменам, которые они обслуживают, а резолверам — обрабатывать только запросы собственных клиентов — например, пользователей корпоративной сети. В реальности, к сожалению, зачастую авторитетные серверы также выполняют функцию резолверов, а резолверы готовы обслужить запрос любого клиента. Такие серверы получили название «открытых резолверов».

Открытые резолверы представляют серьезную проблему, поскольку являются идеальными рефлекторами-усилителями: они готовы обслужить запросы от любого клиента, их число колоссально, они повсеместны.

Решение этой проблемы, а точнее, «закрытие» резолвера обычно состоит в задании списков доступа, в который включаются сети локальных клиентов, для которых данный резолвер обеспечивает трансляцию имен. Рекомендации по обеспечению правильного функционирования резолвера опубликованы в RFC 5358[35]. Однако даже в этом случае при отсутствии мер по антиспуфингу, например, описанных в рекомендации BCP38 «Входная фильтрация: поражение атак отказа в обслуживании, использующих спуфинг IP-адресов источника»[36], остается возможность использования резолвера для атаки локальных клиентов.

---

[35] RFC 5358: Preventing Use of Recursive Nameservers in Reflector Attacks, URL: https://www.rfc-editor.org/rfc/rfc5358

[36] BCP38: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing, URL: https://www.rfc-editor.org/info/bcp38

```
$ dig +dnssec isc.org ANY
;; QUESTION SECTION:
;isc.org.                       IN      ANY

;; ANSWER SECTION:
dig +dnssec isc.org ANY
;; QUESTION SECTION:
;isc.org.                       IN      ANY

;; ANSWER SECTION:
isc.org.                7200    IN      RRSIG   SPF 5 2 7200 20131026
isc.org.                7200    IN      SPF     "v=spf1 a mx ip4:204.15
isc.org.                7200    IN      RRSIG   DNSKEY 5 2 7200 2013
isc.org.                7200    IN      RRSIG   DNSKEY 5 2 7200 2013
isc.org.                7200    IN      DNSKEY  257 3 5 BEAAAAOhHQD
isc.org.                7200    IN      DNSKEY  256 3 5 BQEAAAABwuH
isc.org.                3600    IN      RRSIG   NSEC 5 2 3600 20131026
isc.org.                3600    IN      NSEC    _adsp._domainkey.isc.org.
isc.org.                7200    IN      RRSIG   NAPTR 5 2 7200 20131026
isc.org.                7200    IN      NAPTR   20 0 "S" "SIP+D2U" "" _sip._
isc.org.                60      IN      RRSIG   AAAA 5 2 60 2013102620273
isc.org.                60      IN      AAAA    2001:4f8:0:2::69

…

;; ADDITIONAL SECTION:
asterisk.isc.org.       300     IN      A       149.20.32.15
mx.pao1.isc.org.        3600    IN      A       149.20.64.53
mx.pao1.isc.org.        3600    IN      AAAA    2001:4f8:0:2::2b
mx.ams1.isc.org.        3600    IN      A       199.6.1.65
mx.ams1.isc.org.        3600    IN      AAAA    2001:500:60::65
ams.sns-pb.isc.org.     7200    IN      A       199.6.1.30
ams.sns-pb.isc.org.     7200    IN      AAAA    2001:500:60::30
ord.sns-pb.isc.org.     7200    IN      A       199.6.0.30
ord.sns-pb.isc.org.     7200    IN      AAAA    2001:500:71::30
sfba.sns-pb.isc.org.    7200    IN      A       149.20.64.3
sfba.sns-pb.isc.org.    7200    IN      AAAA    2001:4f8:0:2::19
_sip._udp.isc.org.      7200    IN      RRSIG   SRV 5 4 7200 20131026202736 2013

…

;; Query time: 121 msec
;; SERVER: 199.6.1.30#53(199.6.1.30)
;; WHEN: Fri Sep 27 18:53:51 2013
;; MSG SIZE  rcvd: 3987
```

Рис. 24. Ответ на запрос всех существующих записей имени isc.org с поддержкой DNSSEC.

Response Rate Limiting (RRL) — ограничение частоты ответов

Если в случае резолверов проблему можно решить, оставив доступ только клиентам локальной сети, то для авторитетных серверов такой подход невозможен. По определению, они должны отвечать на запросы любого клиента. В этом случае негативный эффект можно уменьшить, применяя механизм ограничения частоты ответов, или RRL.

Этот механизм[37] был предложен Полом Викси (Paul Vixie) и Верноном Схряйвером (Vernon Schryver) и сначала внедрен в ПО BIND 9 (ISC), а впоследствии Knot DNS (CZ-NIC) и NSD (NLNetLabs).

Идея механизма проста: сервер отвечает только на ограниченное число запросов с идентичным ответом от одного и того же клиента. Ограничение определяется заданным администратором параметром — число ответов в секунду. Идентичными считаются ответы для одного и того же существующего доменного имени (QNAME) и типа записи (QTYPE). Также ответы на несуществующие поддомены (NXDOMAIN) или пустые запросы (NODATA) считаются идентичными и, соответственно, учитываются при подсчете.

Идентичными считаются клиенты, принадлежащие одной и той же сети (адресному блоку), размер которой задается администратором.

Этот механизм в первую очередь предназначен для авторитетных DNS-серверов. В случае рекурсивных резолверов применять его следует с большой осторожностью, чтобы не нарушить работу локальных клиентов. Дело в том, что ввиду недостаточного кеширования DNS-ответов многими приложениями повторяемость одинаковых запросов к резолверам от локальных клиентов достаточно велика, что может включить механизм ограничения. Например, при получении почтового сообщения сервер SMTP сделает запрос на записи NS, PTR, A и AAAA для входящего SMTP-соединения. Далее, при получении команды «Mail From» для этого же сообщения, сервер сделает дополнительные запросы для записей NS, A, AAAA, MX, TXT и SPF. Некоторые веб-браузеры также запрашивают одни и те же имена при обработке встроенных в веб-страницу изображений. Как мы уже говорили, наиболее правильным решением в этом случае является «закрытие» резолвера таким образом, чтобы он отвечал только на запросы, поступающие от локальных клиентов.

Механизм RRL имеет специальную возможность, позволяющую частично отвечать на запросы «нормальных» клиентов, адреса которых используются в попытке рефлекторной атаки. Вместо полного блокирования ответа на каждый второй запрос (этот параметр конфигурируется) сервер отвечает неполным, «обрезанным» ответом — небольшим пакетом с установленным флагом TC (truncation bit). После этого правильно функционирующий клиент должен сделать попытку получить полный ответ с помощью транспортного протокола TCP. Поскольку TCP требует установления соединения (включающего троекратный обмен данными между клиентом и сервером), для подложных адресов такая возможность исключена.

Использование технологии anycast

Когда DNS является объектом атаки, уменьшению рисков может помочь технология anycast (аникаст). В этом случае недостатки использования в DNS протокола UDP становятся преимуществами.

[37] https://kb.isc.org/docs/aa-01000

В октябре 2002 года система корневых серверов DNS подверглась по тем временам крайне массированной атаке DDoS. Объемы трафика достигали 100 Мбит/с, а суммарная мощность атаки превысила 900 Мбит/с. Трафик состоял из пакетов ICMP, TCP SYN, фрагментов TCP и UDP.

В результате атаки, которая продолжалась чуть больше часа, несколько корневых серверов оказались недоступны для большинства клиентов DNS. Интересно, что мощность самих серверов позволила бы им справиться с объемом запросов, но во многих случаях оказалась перегруженной связующая инфраструктура, что и привело к потерям трафика и отказу в обслуживании.

Атака 2002 года, хотя и имела незначительный эффект на производительность глобальной DNS, получила широкое освещение в прессе. Анализ атаки также послужил толчком для серьезного рассмотрения технологии аникаст в качестве способа распределения нагрузки. Аникаст позволяет уменьшить концентрацию трафика и локализовать атаку, создавая местные «точки притяжения».

Также технология аникаст позволяет решить задачу увеличения числа серверов, обслуживающих зону, без увеличения числа записей авторитетных серверов (записей NS) этой зоны, которое имеет свои пределы.

Во-первых, чем больше список серверов, тем больше размер ответа-реферала. Это, кстати, послужило причиной ограничения числа корневых серверов — 13, связанного с максимально допустимым размером сообщения в 512 байт, установленным стандартом DNS [RFC 1035 4.2.1]. Исторически это ограничение было вызвано максимальным размером пакета UDP, гарантирующим отсутствие фрагментации. И хотя сегодня расширение DNS EDNS0 (RFC 2671 2.3, 4.5) предусматривает предварительное соглашение о размере сообщения между клиентом и сервером, типичный размер пакета не превышает 4 кбайт.

Во-вторых, увеличение числа записей NS может негативно влиять на производительность системы. Например, в исследовании, проведенном Джефом Хьюстоном (Geoff Huston) с коллегами[38], было обнаружено, что некоторые резолверы при ошибке валидации DNSSEC повторяют эту попытку для всех существующих авторитетных серверов.

Существенно повысить устойчивость системы может применение технологии аникаст, известной с 1993 года, но ранее не использовавшейся в глобальном масштабе. Ее суть в том, что оператор анонсирует одну и ту же сеть (префикс IP и автономную систему) в различных частях Интернета. Благодаря архитектуре системы маршрутизации для любого клиента существует единственный и самый «короткий» путь к любой другой сети Интернета. Таким образом, аникаст позволяет клиенту установить связь с наиболее близкой в топологи-

[38] https://www.potaroo.net/ispcol/2010-02/rollover.html

ческом смысле сетью без дополнительных изменений в ПО и протоколах! Принцип работы аникаст более подробно изложен в RFC 3258 «Распределение авторитетных серверов имен с использованием общего юникаст-адреса»[39], а схематично показан на рис. 25.

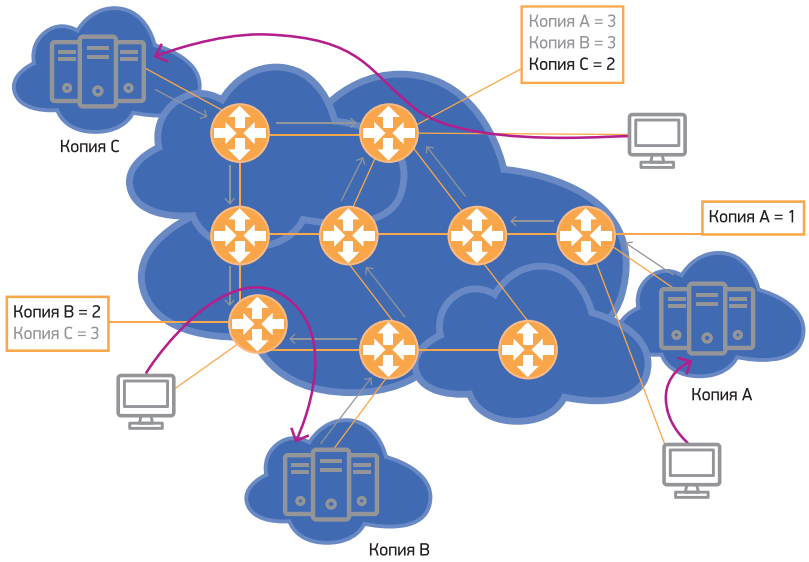

Рис. 25. Использование технологии аникаст для распределения нагрузки и повышения устойчивости DNS. Благодаря правилам выбора «лучшего пути» по протоколу динамической маршрутизации BGP каждый из клиентов получает доступ к «ближайшему» серверу DNS. Например, если «Копия A» и «Копия B» отстоят от клиента на три сетевых сегмента, а «Копия C» — на два, то в общем случае она и будет выбрана при маршрутизации трафика.

Технология аникаст наиболее подходит для приложений, которые используют протокол UDP, работающий без установления продолжительной связи. Например, при использовании TCP при каких-либо изменениях в топологии Интернета (которые происходят постоянно) кратчайший путь может измениться в процессе сеанса и привести клиента к другой сети аникаст, в результате чего связь будет разорвана.

В 2003 году после тщательной экспертной проверки и тестирования эта технология была впервые применена на корневом уровне DNS. Консорциум ISC, оператор корневого сервера f.root-servers.net (о корневых серверах мы более подробно скажем в разделе «Корневой уровень DNS»), разместил

[39] RFC 3258: Distributing Authoritative Name Servers via Shared Unicast Addresses, URL: https://www.rfc-editor.org/rfc/rfc3258

реплику своего сервера с использованием аникаст. Примеру ISC последовал ряд других операторов, и география системы КС существенно расширилась, как можно видеть на рис. 26. За период с 2003 по 2023 год число серверов выросло с изначальных 13 до более 1600.

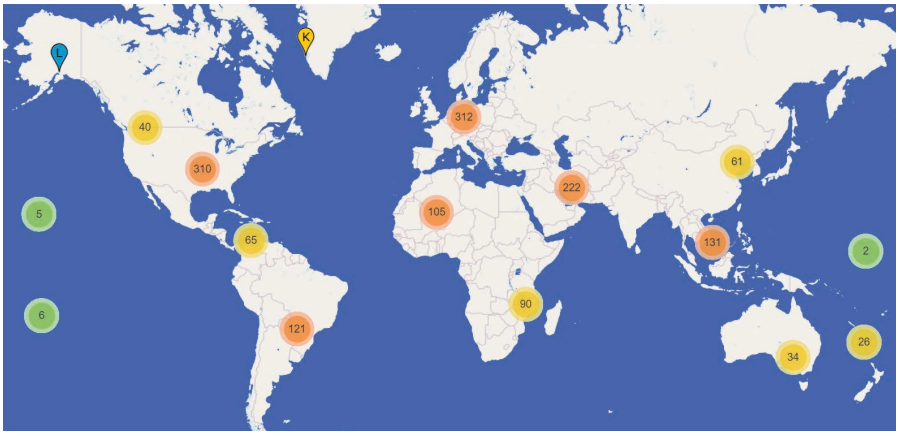

Рис. 26. Карта размещения корневых серверов DNS (апрель 2023 г.).
Источник: https://root-servers.org

Расширения безопасности DNS — DNSSEC

Теперь поговорим более подробно о технологии, позволяющей защитить многие уязвимые места DNS.

В рамках IETF были расширены возможности стандартного протокола DNS для решения проблемы аутентичности и целостности данных. Эти расширения получили название DNSSEC. Основная спецификация DNSSEC содержится в стандартах IETF RFC 4033[40], RFC 4034[41], RFC 4035[42] и RFC 5155[43]

DNSSEC позволяет пользователю убедиться, что полученные данные не были модифицированы в процессе публикации и передачи. Пользователь может быть уверен, что данные, содержащиеся в ответе на запрос DNS, в точности соответствуют данным в зоне для запрашиваемого доменного имени.

---

40 RFC 4033: DNS Security Introduction and Requirements,
URL: https://www.rfc-editor.org/rfc/rfc4033

41 RFC 4034: Resource Records for the DNS Security Extensions,
URL: https://www.rfc-editor.org/rfc/rfc4034

42 RFC 4035: Protocol Modifications for the DNS Security Extensions,
URL: https://www.rfc-editor.org/rfc/rfc4035

43 RFC 5155: DNS Security (DNSSEC) Hashed Authenticated Denial of Existence,
URL: https://www.rfc-editor.org/rfc/rfc5155

В рамках стандартного подхода к информационной безопасности рассматриваются три основных аспекта данных: их конфиденциальность, целостность и доступность. Целью технологии DNSSEC является защита целостности данных, а точнее — обеспечение возможности проверки целостности данных. Кстати, при использовании аникаст вопросы аутентичности и целостности данных встают еще острее, увеличивая ценность DNSSEC.

DNSSEC основан на криптографии с использованием открытых ключей. Администратор зоны подписывает все записи зоны своей цифровой подписью. Там же администратор публикует и открытый ключ, соответствующий этой цифровой подписи. Таким образом, путем проверки подлинности цифровой подписи и ее принадлежности администратору зоны пользователь может убедиться в целостности и аутентичности полученных данных.

Подлинность открытого ключа удостоверяет администратор родительской зоны путем включения в зону так называемой записи DS, представляющей собой хеш открытого ключа дочерней зоны. Разумеется, эта запись заверена цифровой подписью администратора этой родительской зоны. Его ключ, в свою очередь, удостоверяется администратором зоны верхнего уровня — и так далее, пока не будет достигнута так называемая точка доверия (trust anchor) — ключ, которому пользователь абсолютно доверяет. В DNSSEC таким ключом является ключ корневой зоны «.».

Таким образом, для проверки подлинности ответа на запрос DNS пользователю необходимо совершить целую цепочку проверок — от корневого ключа до ключа зоны, содержащей доменное имя. Этот процесс называется построением цепочки доверия. Только при успешном построении цепочки доверия и положительной проверке самой записи ответ может быть положительно удостоверен. Если хотя бы одна из проверок заканчивается неудачей, то и общий результат будет отрицательным.

К счастью, цепочка доверия DNSSEC полностью соответствует структуре делегирования имен, поэтому процесс построения цепочки доверия и разрешения имен происходит параллельно. На рис. 27 показан процесс разрешения имени с использованием DNSSEC.

Кстати, из рисунка видно, что в DNSSEC используются два типа ключей — так называемый ключ KSK (Key Signing Key, ключ подписи ключей) и ключ ZSK (ключ подписи зоны). Первый является ключом долговременного пользования и, соответственно, более сильным в криптографическом плане. Именно хеш ключа KSK «экспортируется» в родительскую зону в виде записи DS, устанавливая тем самым цепочку доверия. Ключ ZSK служит для подписания записей самой зоны. Ключ ZSK удостоверяется администратором домена путем подписи его ключом KSK. Поскольку в изменении ZSK задействован только администратор домена, этот процесс может (и должен) происходить достаточно часто.

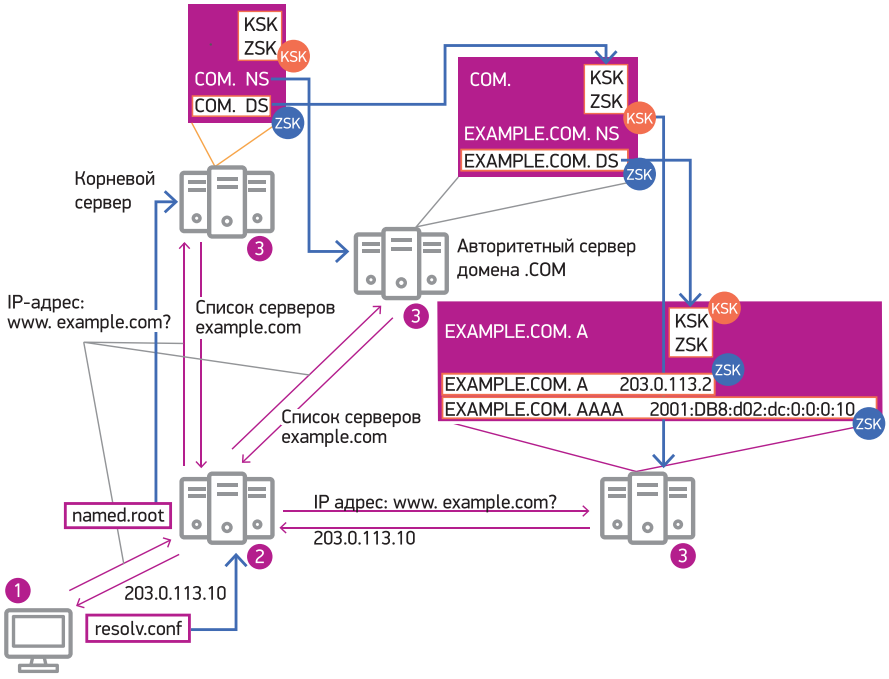

Рис. 27. Дополнительные элементы в процессе разрешения имен при внедрении DNSSEC.

Еще одна важная особенность DNSSEC — он позволяет удостовериться в несуществовании имени. Другими словами, при запросе разрешения несуществующего имени пользователю будет возвращен криптографически заверенный ответ, указывающий на отсутствие запрашиваемой записи в зоне. Для этого в DNSSEC используются записи NSEC и NSEC3[44]. Для того чтобы получить представление о значении различных записей DNSSEC, посмотрите на зону из примера в начале главы, но теперь сгенерированную с использованием DNSSEC (ниже). Как вы видите, размер зоны значительно увеличился за счет появления новых записей: RRSIG, DNSKEY, NSEC.

Запись RRSIG является основным компонентом DNSSEC, поскольку она содержит электронную подпись других записей ресурса, удостоверяющую их подлинность.

Публикация открытых ключей KSK и ZSK осуществляется с помощью записи DNSKEY. Подлинность ключей проверяется с помощью записи DS, которая находится в родительской зоне и соответствует ключу дочерней зоны. Для проверки пар

44 https://ru.wikipedia.org/wiki/DNSSEC

DS–DNSKEY в DNSSEC строится цепочка доверия, заканчивающаяся в корневой зоне (об этом мы подробнее поговорим в разделе «Подписание корневой зоны»).

Наконец, как мы уже говорили выше, записи NSEC и NSEC3 используются для удостоверения «несуществования имени». Для каждого существующего имени в зоне присутствует запись NSEC или NSEC3. Запись NSEC указывает, какие типы записей имеются для данного имени, а также на следующее имя в зоне. Недостатком NSEC является то, что с ее помощью можно легко определить все существующие имена, зарегистрированные в зоне. Чтобы не допустить этого, была разработана другая запись — NSEC3, в которой вместо явного имени используются хеши.

Таблица 4. Зона домена example.com, сгенерированная с использованием DNSSEC

Имя	TTL	Класс	Тип	Значение
$ORIGIN example.com.				
$TTL 86400				
@		IN	SOA	ns1.example.com. hostmaster.example. com. (
				2001062501 ; серийный номер зоны
				21600 ; обновлять каждые 6 ч
				3600 ; следующая попытка после 1 ч
				604800 ; срок годности 1 неделя
				86400) ; минимальный TTL 1 день
		IN	RRSIG	SOA 8 2 86400 20140627093132 20140530093
				132 1443 example.com. (
				bmeJzAuSoYVAEynxoGtogoL710DZ/LrMLTib
				TEqm5rVWLsooTnMhIVb1ySS2paKB5xTskW
				bALou9jzwr6XRvRIEVc1YpYtMWhhEdrKYSG
				PRNCD/ShFi73LBVn7vwdqZL2sV6x+MyjfldT
				64zNl45q56Z4yeSNEH96ni4qA5nVoY=)
		IN	NS	ns1.example.com. ; серверы имен,
				обслуживающие
		IN	NS	ns2.example.com. ; домен example.com
		IN	RRSIG	NS 8 2 86400 20140627093132 201405300931
				32 1443 example.com. (
				kqbwys/cZBo5NbWyg+o5ygLruxk8Xzdg7Czg aVaU-
				nIfiXD3PpUGSDtoDgN5CJHioSoPHC7zz
				WrQ8ep6AZy4WJHySJT9M7Hkzo3CezotiTL
				EdWW24bBgrohmoPzhsshtiAEig/sTP9ywk+
				+U3vfuwWZhWBaAOw6QF3f/WkCM9wdM=)

```
                IN      MX      10 mail1.example. com.          ; почтовые серве-
ры,
                                                                обслуживающие
I               N       MX      20 mail2.example. com.          ; домен
                                                                example.com

                IN      RRSIG   MX 8 2 86400 20140627093132 201405300931
                                32 1443 example.com. (

                                CTPgHC6fYu9wqMZn51r2FpmEO3wcx721rrG
                                9taLM4P5VZrZPHOAk2JiYFkLFGrDt21ecI6U
                                sr9onIDwc/00/qKZI6pzPQA3p31YER030Sopu
                                OIghVDOpgrI4AoleiyK5hQPlpqP7OoxDp3WH
                                MwI8U4JKr8Eqw8RkbAfvIiRtF9k=)


        3600    IN      DNSKEY  257 3 8 (              ; KSK

                                AwEAAb4WMOTBLTFvmBra5m6SK4VfViOz
                                mvyUAUoqv861ZQXeEFvwlndqNU9rwRsMx
                                rSWAYs5nHErKDn49usC/HyxxW1477iGFHh
                                fgL4mjNreJm9z  2QFB1VLbRbEPYdDMLCn
                                4cooqnG7/KG8W2i8Pym1L7f+aREwbLo+/71
                                6AS2PbaKMhfWLKLiq5wnBcUClQMNzCiwh
                                qxDJp1oePq
                                fkVdeUgXOtgiodYRIKyQFhJ5VWJ22npoi/ GifoXLCA-
                                DAIAwRLKc8o/yJkCxskzgp

                                Hpw5Cki1Iclgoaq4ssOuPRQ+ne6IHYCz9D2 mwzul-
                                blhLFamKdq7aHzNt4NlyxhpANVFi KLD8=
                                ); key id = 29332, size = 2048b 3600
                IN      DNSKEY  256 3 8 (              ; ZSK

                                AwEAAbd9WqjzE2Pynz21OG5doSf9hFzMr5d
                                hzz2waZ3vTa+005r7AjTAqmA1yH/B3+aAMih
                                Um5ucZSfVq07+kOaRE8yFj9aivOmA1n1+JLe
                                vJq/oyvQyjxQN2Qb89LyaNUT5oKZliL+uyyh
                                NW3KDR3SSbQ/GBwQNDHVcZi+JDR3RCor7
                                ); key id = 1443, size = 1024b
                IN      RRSIG   DNSKEY 8 2 3600 20140627093132 20140530
                                093132 29332 example.com. (

                                TozkmKFrlkoHt9y8vmioO12tWb1LhKsC64C3H
                                6+iDmsoQBYF9VDGYonDePw9lO4hZAWGaVN
                                fpEnrRQ+TRJrvyR5zNfxwca8D86jZ6ydKxpPll
                                2YshJxktsFcfdW6wdkaEhfwXSnkcIyYquIAQE
```

```
                                    B9TgmPoqTkFbgqLqW/4msbYQFcHqSetmoLju
                                    F+yqKoZnpWSt6eCodsGJnLzi3r1CcBytGSouz
                                    vjbhwZo3dhqEbWrY8MVg1H2Ybm1hAtcpOOqij
                                    a2hcF4rTyDxMau7RIzTGyFLLx1EoqtiOKHWbq
                                    2zzxj3VrEQZje4Uwimf9YSctTXk203km+Py2D
                                    WZWRWcoUaFQA==)

        86400       IN      NSEC    ftp.example.com. NS SOA MX RRSIG NSEC DNSKEY
                    IN      RRSIG   NSEC 8 2 86400 20140627093132 20140530093
                                    132 1443 example.com. (
                                    fryQNKmBGDtlPomAwzgoPfRPFmTDoL+zAs9
                                    d7BHrSk/y9uzmp9bPNS4Nob5np4vVbZOSH2
                                    TgpEDon8vjUyPAnOZBZ8D9/riq6UQ6+EMUhD
                                    zZ7PC1j323uLiLF2VuBznADuNGd2sjEZmLUi9 YToZ-
                                    ZIAGEMpZNfOBUQW2llhrC1uI=)
            ftp     IN      A       203.0.113.12
                    IN      A       203.0.113.13
                    IN      RRSIG   A 8 3 86400 20140627093132 20140530093132
                                    1443 example.com. (
                                    mqL5M7SkyiLCDQFFQPcZB/P2zoq6Ezf9x51z/
                                    LQuddtVldoliEA4C1quLh2rMwQwcPSuiReUct
                                    SQ4LudKGsmB6noPTOUEmYYLT7nnJaMcqrm
                                    2mGp5xUruGKPtkboaGjsabQFai/1/Stofz8EyR
                                    ytfoJHMOGdUe+1xgPl7F4yVJQ=)
                    IN      NSEC    mail1.example.com. A RRSIG NSEC
                    IN      RRSIG   NSEC 8 3 86400 20140627093132 20140530093
                                    132 1443 example.com. (
                                    fd1h4zINeX+eAyRdTTIWyzZxn1DSXphhNUpb
                                    Cllig6bljdhfMiKyVVz45f55q+GA8gB43+hBJ5j
                                    2gj7bt3ytfYHi3Jp8ZWSTQUtTo+8NPfrBVunZ Yx-
                                    sNWEerBAyVK8ZecVSjDq+87fyGbQjjX6oU kxai-
                                    guVpFI97/6cKVKQ2ePo=)

mail1               IN      CNAME   server1
...
```

В то же время DNSSEC не является панацеей. Даже если полученные адреса серверов являются правильными, обмен данными может быть перехвачен или перенаправлен с использованием уязвимых мест системы маршрутизации (подробнее — в статье автора «Безопасность системы маршрутизации Интернета»). Также DNSSEC не обеспечивает шифрование данных, здесь необходима другая, довольно широко распространенная технология TLS (Transport Layer Security), которая использует цифровые сертификаты X.509. Именно на технологии TLS базируется протокол HTTPS. Наконец, DNSSEC не спасет от го-

мографии, когда имя сервера внешне очень похоже на другое имя, но на самом деле использует другие символы: например, цифра «1» и буква «l».

И все же использование DNSSEC совместно с другими средствами защиты существенно усиливает их эффективность. Например, в противоположность сертификатам доменных имен, используемых в TLS/HTTPS, цепь доверия в DNSSEC следует цепочке делегирования доменов и, таким образом, основана на деловых отношениях, существующих при регистрации доменов. Об этом мы более подробно поговорим в разделе «Усиление безопасности других услуг с помощью DNS».

Вопросы внедрения DNSSEC

Внедрение DNSSEC является комплексной задачей, требующей участия многих сторон — администраторов и операторов доменных зон и в первую очередь национальных доменов, а также регистраторов, провайдеров хостинга, сетевых провайдеров и разработчиков программного обеспечения. Подобно другим новым технологиям, преимущество использования DNSSEC зависит от степени ее внедрения. Увы, порой образуется замкнутый круг — держатели доменных имен не видят смысла инвестиций во внедрение DNSSEC, пока значительная часть клиентов не станет осуществлять валидацию. А клиенты, в свою очередь, ожидают более масштабного распространения DNSSEC в дереве DNS.

Дополнительные затраты

Да, применение DNSSEC имеет значительные преимущества, но с внедрением этой технологии связаны дополнительные затраты и трудности.

Увеличение размера файла-зоны. Как мы уже видели, использование DNSSEC требует создания дополнительных записей в зоне. Основной вклад в размер зоны вносят записи NSEC (NSEC3) для проверки несуществования имени и запись RRSIG, являющаяся цифровой подписью других записей. Увеличение размера зоны варьируется в зависимости от ее содержимого, размеров используемых ключей, метода подписания, но может превышать восемь крат.

Увеличение размера ответов. Поскольку каждый ответ содержит цифровую подпись, размер ответов также увеличивается в несколько раз. Большой размер ответов может также усилить ущерб от возможной DDoS-атаки, когда множество небольших запросов DNS, посланных с использованием подложного IP-адреса жертвы (и якобы от его имени), вызывают ответный трафик, значительно превышающий входящий. Об этом мы поговорим в следующем разделе.

Увеличение числа запросов. По мере внедрения DNSSEC можно ожидать некоторое увеличение числа дополнительных запросов на получение ключей и построение цепочек доверия.

Большая нагрузка на «клиента». Помимо разрешения имен, клиенту необходимо производить проверку подписей, осуществлять построение цепочек доверия.

Потребуется дополнительная вычислительная мощность, хотя при сегодняшнем развитии технологии это вряд ли является существенной проблемой.

Усложнение инфраструктуры и процесса генерирования зоны. Внедрение DNSSEC требует дополнительной инфраструктуры для генерирования, хранения и обновления ключей, подписания зоны, а также для связанных с этими операциями технических и административных процессов. В дополнение DNSSEC вводит в DNS понятие абсолютного времени, за точностью которого необходимо следить.

Применение DNSSEC на пользовательском уровне

DNSSEC является достаточно зрелой технологией, внедренной в значительном числе доменов верхнего уровня (на июль 2023 года - более 90% всех доменов верхнего уровня[45]). Тем не менее, использование DNSSEC на пользовательском уровне пока незначительно. А ведь именно это определяет эффективность защиты, которую обеспечивает технология DNSSEC.

Существует несколько задач, решение которых необходимо для более широкого распространения DNSSEC на пользовательском уровне.

Внедрение DNSSEC для доменных имен второго и третьего уровня

Если для доменов верхнего уровня ситуация более или менее благополучна, то среди доменных имен второго и последующих уровней DNSSEC практически не внедрен. Очевидно, что для получения эффекта от этой технологии DNSSEC должен быть поддержан по всей цепочке доверия — от собственно запрашиваемого доменного имени до корня DNS.

Подписание корневой зоны явилось хорошим стимулом для операторов национальных и других доменов верхнего уровня. В свою очередь, подписание национального домена и поддержка безопасной делегации — то есть записей DS, удостоверяющих открытые ключи дочерних зон, — стимулирует держателей доменов нижних уровней к внедрению DNSSEC.

Однако этого все же недостаточно. Необходима также административная поддержка DNSSEC на уровне регистраторов. Как минимум нужно обеспечить поддержку создания записи DS в родительской зоне. Но для практического применения эта процедура должна быть существенно упрощена, как, например, это делают GoDaddy[46] и Cloudflare[47].

Для широкомасштабного внедрения DNSSEC также необходима поддержка этой технологии провайдерами хостинга, особенно если последние не являются и регистраторами доменных имен. Поскольку клиентами провайдеров, как

[45] https://rick.eng.br/dnssecstat
[46] https://uk.godaddy.com/help/turn-dnssec-on-or-off-6420
[47] https://developers.cloudflare.com/dns/dnssec

правило, являются простые пользователи, процедура включения DNSSEC должна быть максимально упрощена и автоматизирована, включая создание и обслуживание ключей, подписание и безопасное делегирование. Например, шведский хостинг-провайдер Binero[48] автоматически подписывает все домены, хостинг которых он предоставляет. Таким образом, для включения DNSSEC пользователю не приходится делать практически ничего.

Поддержка DNSSEC резолверами сервис-провайдеров и корпоративных сетей

Как бы широко ни был внедрен DNSSEC в системе DNS, без проверки подлинности ответов на пользовательском уровне эффект от этой технологии сводится к нулю. Данная задача может быть решена поэтапно.

Наиболее простой и в то же время эффективный шаг — обеспечение поддержки DNSSEC резолверами Интернета сервис-провайдеров и корпоративных сетей. Поскольку разрешение имен для пользователей сетей доступа и корпоративных сетей осуществляют именно эти резолверы, проверка подлинности ответов позволит усилить защиту DNS для широкой клиентской базы. Безусловно, слабым звеном в этой схеме остается сегмент сети между резолвером и пользовательским приложением (например, веб-браузером и почтовым приложением), но, как правило, этот сегмент находится в зоне административного контроля провайдера и может быть эффективно защищен от потенциальных угроз, которые рассматривались выше. Примером такого подхода является американский провайдер широкополосного доступа Comcast[49], внедривший в январе 2012 года DNSSEC в своей инфраструктуре резолверов, что обеспечило защиту DNSSEC для 17,8 миллиона пользователей. Большинство сетевых сервис-провайдеров Швеции также осуществляют проверку имен для своих пользователей.

По состоянию на июль 2023 года, согласно статистике APNIC Labs[50], более 30% пользователей в мире производят валидацию ответов с помощью DNSSEC. Значительный вклад здесь вносит «публичный» резолвер Google — Public DNS, который обслуживает 14% запросов с DNSSEC. Как видно из рис. 28, процент проникновения DNSSEC на пользовательском уровне существенно различается для стран и сетей.

Поддержка DNSSEC на уровне приложений

Полноценной моделью «сквозного» внедрения DNSSEC, которая может рассматриваться как следующий шаг, является проверка подлинности ответа самим приложением или операционной системой. При этом приложение может по-прежнему производить разрешение имени через резолвер сервис-провайдера, но последний вместо проверки подлинности полученных

[48] https://binero.com
[49] http://www.comcast.com
[50] https://stats.labs.apnic.net/DNSSEC

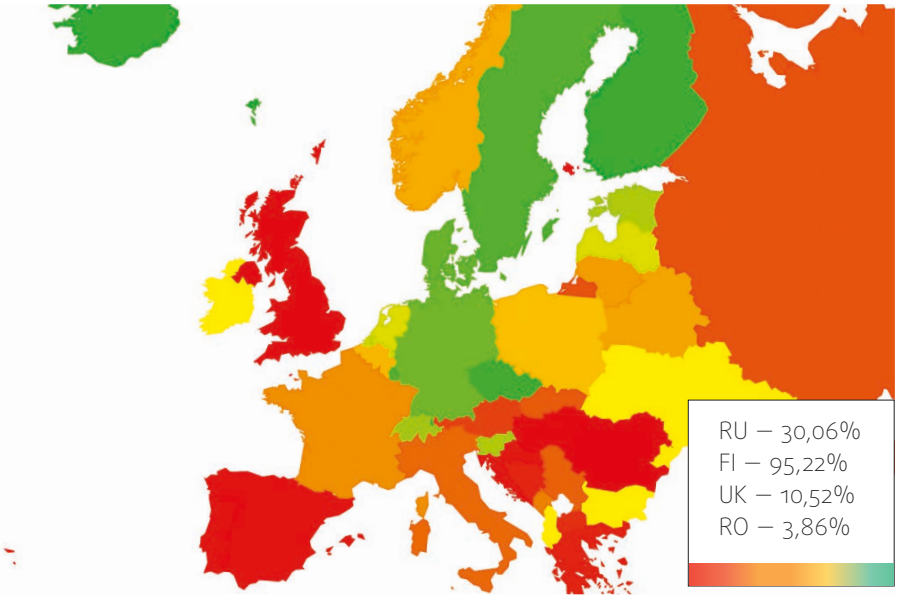

Рис. 28. Карта использования DNSSEC европейскими пользователями Сети. (данные на февраль 2024 г.)
Источник: https://stats.labs.apnic.net/dnssec/XE

ответов и выдачи окончательного результата будет передавать «сырые» данные (то есть включающие записи RRSIG, NSEC/NSEC3 и DNSKEY) непосредственно приложению. Таким образом задача итеративного разрешения имени и построения цепочки доверия ложится на пользовательское приложение (или операционную систему), а резолвер используется в целях кеширования и оптимизации трафика. Примером реализации такого подхода является программа DNSSEC-Trigger[51], работающая совместно с установленным на компьютере пользователя локальным итеративным резолвером Unbound[52] от компании NLnet Labs.

Защита конфиденциальности в DNS

При разработке DNS основным требованием с точки зрения безопасности была доступность. Позже стало очевидно, что целостность данных DNS очень важна, и для защиты данных DNS при передаче были разработаны стандарты DNSSEC. Но до недавнего времени достижение конфиденциальности никогда не было целью. В конце концов, данные DNS общедоступны по определению.

Однако, хотя сами данные DNS являются общедоступными, связанные с ними метаданные и, в частности, данные о том, кто и что именно запрашивает, рассматриваются как серьезная угроза конфиденциальности пользователей.

[51] https://nlnetlabs.nl/projects/dnssec-trigger/about
[52] https://nlnetlabs.nl/projects/unbound/about

Существует два основных подхода к решению проблем конфиденциальности DNS. Во-первых, затруднить прослушивание DNS-запросов с помощью шифрования DNS-транзакций. Это поддерживают два основных механизма — DNS-over-TLS (DoT) и DNS-over-HTTP (DoH). Второй подход заключается в уменьшении раскрытия информации за счет уменьшения объема информации в каждом DNS-запросе.

Эти механизмы предоставляют пользователю более высокий уровень защищенности частной информации, поскольку клиентские DNS-запросы становятся невидимыми для пассивного наблюдателя. Однако стоит помнить, что вероятность утечки все же есть, так как ваши запросы и ответы на них известны не только вам. В системе, где между вами и авторитетными DNS-серверами есть посредники (резолверы), абсолютной конфиденциальности быть не может. Даже если вы отправляете запросы на авторитетные DNS-серверы напрямую, они все равно будут знать, что вы запросили у них эту информацию. Поэтому имеет смысл внимательно ознакомиться с политикой конфиденциальности вашего поставщика услуг DNS, прежде чем сделать выбор.

DNS поверх TLS

DNS поверх TLS (DoT) задокументирован в RFC 7858[53]. Он использует протокол Transport Layer Security (TLS) для шифрования связи между клиентом и сервером, а также для того, чтобы клиент мог аутентифицировать сервер. Во многом так же, как TLS используется для защиты сеансов HTTP и обеспечения некоторой уверенности в том, что сервер является авторизованным агентом указанной службы, этот протокол также может использоваться в контексте DNS между DNS-клиентом (обычно это резолвер-заглушка операционной системы) и выбранным рекурсивным резолвером, поддерживающим DoT.

DNS поверх HTTPS

DNS поверх HTTPS (DoH) — это стандарт, задокументированный в RFC 8484[54]. Он похож на DoT тем, что позволяет шифровать все DNS-запросы, так что только DNS-клиент (обычно это браузер пользователя) и сервер DoH по вашему выбору (обычно рекурсивный резолвер, поддерживающий DoH) знают, на какие сайты собирается перейти пользователь. Описанный подход — это больше, чем туннель через HTTP. Он устанавливает типы форматирования мультимедиа по умолчанию для запросов и ответов, но использует обычные механизмы согласования содержимого HTTP для выбора альтернатив, которые клиенты могут предпочесть в будущих вариантах использования. В дополнение к этому согласованию типа мультимедиа он согласуется с функциями HTTP, такими как кеширование, перенаправление, проксирование, аутентификация и сжатие.

---

53 RFC 7858: Specification for DNS over Transport Layer Security (TLS), URL: https://www.rfc-editor.org/rfc/rfc7858

54 RFC 8484: DNS Queries over HTTPS (DoH), URL: https://www.rfc-editor.org/rfc/rfc8484

Хотя DoH достигает цели защиты данных пользовательских запросов от прослушивания третьими сторонами, его внедрение вызвало озабоченность, в основном связанную с моделью развертывания, а не с самим протоколом.

Во-первых, DoH очень сложно обнаружить. Он выглядит как HTTPS-трафик и использует тот же порт, что и HTTPS-трафик. Возможность обнаружения DoH путем проверки имени сервера, которое передается в открытом виде во время «рукопожатия» TLS, исчезает с работой над зашифрованным SNI в TLS 1.3. Это ломает многие существующие реализации, от родительского контроля до инструментов, развернутых интернет-провайдерами для соблюдения различных правил. Но опять же — эта проблема возникает только в определенных сценариях развертывания. Например, если резолвер DNS, выбранный пользовательскими приложениями, является резолвером DNS интернет-провайдера, который поддерживает DoH, данные DNS видны интернет-провайдеру и DoH не представляет никакой проблемы.

Это приводит ко второй проблеме — выбору сервера DoH. Вместо использования локально настроенной службы резолвера DNS, предоставляемой интернет-провайдером (например, через запрос DHCP), текущие реализации предписывают использовать службу, настроенную браузером. Например, конфигурация DoH в Firefox предоставляет резолвер DNS Cloudflare по умолчанию, но позволяет пользователю самостоятельно выбрать доверенный рекурсивный резолвер.

DNS поверх QUIC

Хотя DoT и DoH обеспечивают защищённость данных между DNS-клиентом и рекурсивным резолвером, оба протокола используют TCP в качестве транспорта. Помимо значительных накладных расходов (по сравнению с UDP, основным транспортным протоколом DNS), TCP может являться причиной низкой производительности, особенно когда используется многопоточная передача (как в случае HTTPS/2). Дело в том, что при потере сегмента данных TCP будет ожидать повторной передачи этого сегмента, тем самым блокируя получение остальных данных, возможно, относящихся к другому логическому потоку.

Решением этой проблемы, в частности, для HTTPS стал стандартизованный в 2021 году протокол QUIC[55]. Хотя этот протокол был разработан в основном для поддержки HTTPS, QUIC является транспортным протоколом общего применения, и DNS поверх QUIC, или DoQ[56], тому свидетельство.

Основным преимуществом использования QUIC по сравнению с TCP является то, что он использует UDP в качестве базового транспорта, обеспечивая целостность передаваемых данных и мультиплексирование на уровне приложений. Это значительно повышает производительность QUIC, и как следствие – DoQ.

---

55 RFC 9000: A UDP-Based Multiplexed and Secure Transport,
URL: https://www.rfc-editor.org/rfc/rfc9000

56 RFC 9250: DNS over Dedicated QUIC Connections,
URL: https://www.rfc-editor.org/rfc/rfc9250

Минимизация имен Qname

В настоящее время, когда преобразователь получает запрос «Какова запись AAAA для www.example.com?», он отправляет тот же вопрос на все авторитетные серверы, начиная с корневых серверов (см. рис 18. Процесс трансляции имен в DNS). Но отправка полного имени запроса (www.example.com) авторитетному серверу имен является традицией, а не требованием протокола.

В рамках рабочей группы DNSOP IETF был разработан подход, стандартизованный в RFC 9156 «Минимизация имен запросов DNS для улучшения конфиденциальности»[57]. Идея состоит в том, что каждый авторитетный сервер получает только ту часть общего вопроса, на которую он уполномочен отвечать. Например, от корневых серверов преобразователь ожидает узнать имя авторитетных серверов для домена .com, поэтому вместо отправки вопроса «www.example.com.» достаточно запросить информацию (имена авторитетных серверов) о домене «com.».

Усиление безопасности других услуг с помощью DNS

DANE (DNS-based Authentication of Named Entities — Аутентификация поименованных объектов с использованием DNS)

Как мы уже заметили, внедрение DNSSEC также открывает новые возможности. Например, решение проблем, связанных с сертификатами открытых ключей (X.509), используемых для проверки достоверности веб- или почтовых серверов и обмена информацией с использованием защищенного протокола TLS. Эта система обобщенно называется Web-PKI.

Дело в том, что выдачей этих X.509 сертификатов занимаются несколько сотен независимых удостоверяющих центров (УЦ), которые используют данные третьих сторон (например, данные от компаний-регистраторов) для проверки прав пользования доменным именем. Конкретный список доверенных УЦ определяется производителями веб-браузеров – такими компаниями, как Google, Apple, Mozilla Foundation и т.д. Хотя в выборе УЦ они руководствуются требованиями CA/Browser Forum[58], а также результатами аудита WebTrust Task Force[59] и ETSI[60], в основном этот процесс закрыт и непрозрачен.

В Web-PKI все УЦ, корневые сертификаты которых установлены в браузере пользователя, имеют право выдавать сертификаты для любого доменного имени и при этом имеют одинаковый уровень доверия. Любой из этих корневых сертификатов является так называемой точкой доверия (Trust Anchor, TA). Это значит, что УЦ с

---

57 RFC 9156: DNS Query Name Minimisation to Improve Privacy, URL: https://www.rfc-editor.org/rfc/rfc9156

58 https://cabforum.org/about-the-baseline-requirements

59 https://www.cpacanada.ca/en/business-and-accounting-resources/audit-and-assurance/overview-of-webtrust-services/principles-and-criteria

60 https://portal.etsi.org/TB-SiteMap/ESI/Trust-Service-Providers

недостаточным качеством проверок прав владения именем является «слабым звеном» всей системы. Злоумышленник может использовать такой УЦ для получения конкурирующего сертификата для уже существующего чужого имени или сертификата для несуществующего имени. Даже флагманы индустрии оказываются уязвимы – так, компания Symantec на протяжении нескольких лет генерировала сертификаты для несуществующих имен, с просроченным сроком действия и другими нарушениями требований CA/Browser Forum[61].

Дело усугубляется еще и тем, что корневые УЦ часто делегируют полномочия выдачи сертификатов подчиненным УЦ. В результате в Интернете существует большое количество доверенных УЦ, качество регистрационных процессов которых проверить практически невозможно.

Исправить ситуацию можно, если владелец доменного имени будет сам контролировать выдачу сертификатов, а по возможности вообще не будет пользоваться услугами третьих лиц.

Здесь на помощь приходит DNS. Действительно, если владелец имени или доменной зоны контролирует публикацию такой важной информации, как IP-адреса сайтов, почтовых серверов, обслуживающих домен, и т.п., почему бы не создать новую запись, с помощью которой владелец смог бы указать на сертификат, который следует использовать при обращении к сайту или почтовому серверу, связанным с этим доменным именем? Разумеется, запись должна быть защищена, и DNSSEC выполняет эту функцию.

Разработка и стандартизация соответствующих протоколов была выполнена в рамках рабочей группы IETF DANE[62]. DANE позволяет владельцу домена указать, какой сертификат TLS/SSL должно использовать приложение или служба для подключения к вашему сайту. Для этого используется запись TLSA, стандартизованная в RFC 6698[63] и RFC 7671[64].

Основой DANE является новая запись ресурса TLSA. Эта запись с доменным именем хоста содержит инструкции по использованию TLS/SSL-сертификата при доступе к сервису по указанному порту и протоколу. Так, например, запись ниже указывает, что при валидации сертификата, полученного при обращении к веб-серверу www.example.com по протоколу HTTPS (порт 443, протокол TCP), необходимо использовать точку доверия (trust anchor, TA), указанную в последнем поле записи:

---

61 https://wiki.mozilla.org/CA/Symantec_Issues

62 https://datatracker.ietf.org/wg/dane/documents/

63 RFC 6698: The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA, URL: https://www.rfc-editor.org/rfc/rfc6698

64 RFC 7671: The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance, URL: https://www.rfc-editor.org/rfc/rfc7671

_443._tcp.www.example.com. IN TLSA 2 0 1 (E8B54E0B4BAA815B06D3462D65FBC7C0CF5 56ECCF9F5303EBFBB77D022F834C0)

Поля после метки TLSA имеют следующее значение:

IN TLSA	2	0	1	E8B54E0B4BAA8...
	Использование сертификата	Селектор	Тип сравнения	Данные, ассоциированные с сертификатом
	0 – PKIX-TA 1 – PKIX-EE 2 – DANE-TA 3 – DANE-EE	0 – использовать весь сертификат для сравнения 1 – использовать только открытый ключ для сравнения	0 – все данные, указанные селектором, используются для сравнения 1 – хеш SHA-256 данных используются для сравнения 2 хеш SHA-512 данных используются для сравнения	Данные, необходимые для сравнения, закодированные в виде строки BASE64

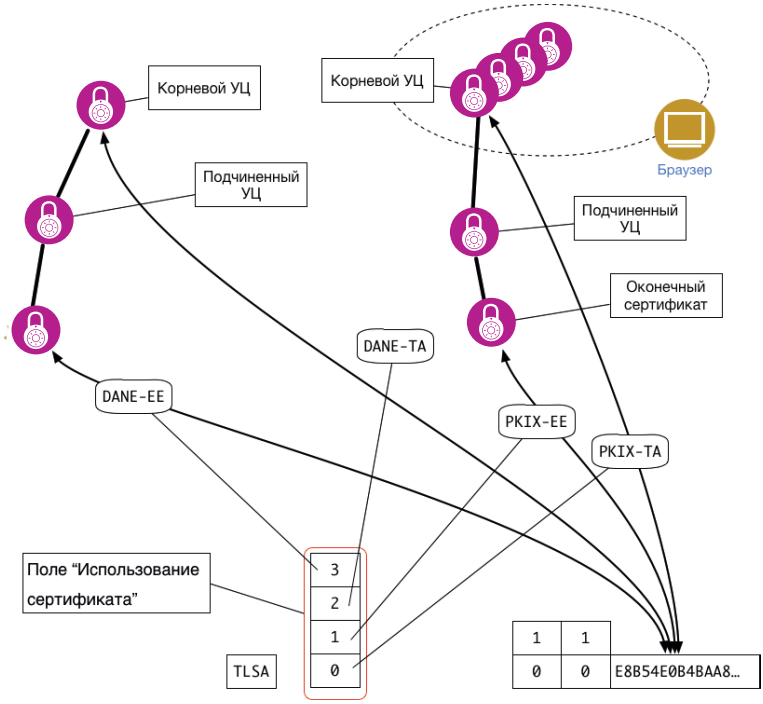

Рис. 29. Различные типы записи TLSA и связанные с ними проверки.

Значение 0 (PKIX-TA) указывает, что путь валидации полученного сертификата (или открытого ключа в зависимости от значения поля «Селектор») должен проходить через сертификат, указанный в поле «Данные, ассоциированные с сертификатом». Это может быть как корневой, так и подчиненный сертификат, но в любом случае путь валидации должен заканчиваться точкой доверия, зарегистрированной в пользовательском браузере.

Значение 1 (PKIX-EE) накладывает ограничение на сертификат, полученный от веб-сервера. Он (или его открытый ключ) должен соответствовать «данным, ассоциированным с сертификатом». При этом сертификат должен пройти проверку, используя путь валидации к одной из точек доверия браузера.

Значение 2 (DANE-TA) применяется для определения сертификата, который должен использоваться в качестве новой точки доверия при валидации сертификата, полученного от веб-сервера. Этот метод также называют «утверждением точки доверия», поскольку он позволяет владельцу доменного имени указать точку доверия, которая не входит в стандартную коллекцию TA пользовательского браузера.

Наконец, значение 3 (DANE-EE) определяет метод, когда сертификат (или его открытый ключ), полученный от сервера, должен совпадать с сертификатом, указанным в записи TLSA. Этот метод позволяет владельцу доменного имени использовать собственные сертификаты без привлечения сторонних УЦ. Этот метод отличается от PKIX-EE тем, что он не требует дополнительной валидации сертификата через путь доверия.

DANE может использоваться не только при взаимодействии с веб-серверами. Например, защита передачи данных между почтовыми серверами является чрезвычайно важной. При этом часто используется протокол TLS, но аутентичность сертификата сервера представляет еще большую проблему, чем в случае с Web-PKI. DANE и здесь окажет администраторам почтовых серверов неоценимую услугу, делая почтовую систему в целом более защищенной. Более подробно о проблематике защиты передачи почтовых сообщений между транспортными почтовыми агентами (Mail Transport Agent, MTA) и о применении DANE в этом случае для защиты протокола SMTP описано в RFC 7672[65].

Другие протоколы приложений, например, IMAP или XMPP, могут также использовать DANE. Обсуждение использования DANE в этих случаях можно найти в RFC 7673[66].

---

[65] RFC 7672: SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS), URL: https://www.rfc-editor.org/rfc/rfc7672

[66] RFC 7673: Using DNS-Based Authentication of Named Entities (DANE) TLSA Records with SRV Records, URL: https://www.rfc-editor.org/rfc/rfc7673

Другими словами, DNSSEC+DANE обладают существенным потенциалом обеспечения защищенности коммуникационной инфраструктуры Интернета на уровне приложений.

DKIM, SPF и DMARC

Одна из проблем существующей инфраструктуры электронной почты заключается в том, что отправитель может использовать любое доменное имя для различных идентификаторов в заголовке почты, а не только собственное доменное имя. Эта уязвимость используется спамерами, чтобы ввести пользователей в заблуждение, подменив домен отправителя. Sender Policy Framework (SPF, Основа политики отправителя)[67] — стандартизированный метод предотвращения подделки адреса отправителя. SPF позволяет администраторам указать, каким хостам разрешено отправлять почту от имени данного домена, путем создания специальной записи SPF в DNS. Почтовые обменники используют эту запись DNS для проверки того, что почта с данного домена отправляется хостом, санкционированным администратором этого домена.

Хотя SPF гарантирует, что почта может поступать только с авторизованных почтовых серверов, он не защищает само сообщение электронной почты, включая его заголовки, такие как «От:», «Кому:», «Дата:». Это означает, что спамер все еще может выдать себя за кого-то, подделав заголовок «От:», и это то, на что пользователь обратит внимание. DomainKeys Identified Mail (DKIM, Идентификация почты с помощью DomainKeys)[68] использует асимметричную криптографию для цифровой подписи сообщения. DKIM возьмет хеш нескольких полей электронной почты, в том числе «От:», «Кому:», «Дата:». Затем этот хеш подписывается закрытым ключом, который генерируется администратором домена и помещается в заголовок DKIM. Открытый ключ домена публикуется в DNS для этого домена и используется для проверки подлинности электронной почты

DKIM не защищает от подделки поля «От:» напрямую, но гарантирует, что электронное письмо действительно пришло из рассматриваемого домена. Например, DKIM может гарантировать, что электронное письмо пришло из домена example.com, но не обязательно может гарантировать, от кого в этом домене отправлено сообщение, поскольку ключ используется для всего домена, а не для отдельных отправителей.

Domain-based Message Authentication, Reporting and Conformance (DMARC, Аутентификация, отчетность и определение соответствия сообщений на основе доменного имени)[69] позволяет владельцу домена публиковать политики обработки сообщений

---

67 RFC 7208: Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1, URL: https://www.rfc-editor.org/rfc/rfc7208

68 RFC 6376: DomainKeys Identified Mail (DKIM) Signatures, URL: https://www.rfc-editor.org/rfc/rfc6376

69 RFC 7489: Domain-based Message Authentication, Reporting, and Conformance (DMARC), URL: https://www.rfc-editor.org/rfc/rfc7489

для получателей сообщений электронной почты, исходящих из этого домена, и запрашивать отчеты об аутентификации полученной почты. Политика DMARC позволяет отправителю указать, что его сообщения защищены с помощью SPF и/или DKIM, и сообщает получателю, что делать, если ни один из этих методов проверки подлинности не проходит – например, поместить сообщение в карантин или отклонить. DMARC также позволяет получателю электронной почты сообщать отправителю о сообщениях, прошедших и/или не прошедших оценку DMARC.

Эти политики публикуются в DNS в виде записей TXT.

Координация и администрирование доменных имен верхнего уровня

Общий взгляд на систему. Структуры ICANN

Координацию глобальной системы имен, а точнее, корневого уровня DNS, осуществляет ICANN (Internet Corporation for Assigned Names and Numbers) — частная некоммерческая корпорация, зарегистрированная в штате Калифорния, США. Задачу координации корневого уровня DNS можно разделить на две части: «что» может быть включено в корневую зону в качестве имени, а также «как» — процедурные и операционные вопросы включения и обслуживания этого имени. Первый вопрос, «что», принадлежит уровню разработки политик и правил. Второй же, «как», — вопрос исполнения принятых политик и правил, внесения изменений в корневую зону и ее обслуживания.

Организационные структуры, созданные для решения этих вопросов, различны.

В структуре ICANN существуют две так называемые организации поддержки — организация поддержки общих имен gNSO (Generic Names Supporting Organization) и организация поддержки национальных доменных имен ccNSO (Country Code Names Supporting Organisation). Они занимаются процедурными аспектами и обеспечивают разработку политик.

Разработка политик gNSO происходит согласно утвержденному процессу разработки политик (Policy Development Process, PDP). Процесс предусматривает работу над определением проблемы, а при разработке решения опирается на широкую поддержку общественности и обратную связь, также используя ресурсы консультативных комитетов: ALAC, GAC, RSAC и SSAC. Принятие политик происходит на основе консенсуса, а ратификация производится Советом ICANN.

Работу gNSO координирует совет, в состав которого входят 22 члена, номинированные так называемыми заинтересованными группами (также именуемые стейкхолдерами, от английского stakeholder) — реестрами, регистраторами, коммерческими и некоммерческими группами. Такая широкая избирательная база позволяет сбалансированно отразить интересы различных сторон.

Чтобы получить представление о типе вопросов, рассматриваемых gNSO, достаточно посмотреть на некоторые принятые политики— «Единая политика разрешения споров о доменных именах», «Политика удаления доменов с истекшим сроком действия», «Политика изменения регистраторов».[70]

Как следует из названия, ccNSO занимается вопросами национальных доменов. Некоторые из них весьма щепетильны, поскольку граничат с национальными интересами государств. Хотя PDP ccNSO по структуре похож на PDP gNSO, участие комитета GAC (Government Advisory Committee, Правительственный консультативный комитет) в нем прописано более явно.

Процесс делегирования и ре-делегирования национальных доменов верхнего уровня описан в документе IANA «Delegating or redelegating a country-code top-level domain (ccTLD)»[71]. Основным методом определения допустимости того или иного доменного имени является включение соответствующего «alpha-2» кода таблицы ISO 3166-1. Другим методом является утверждение имени через так называемый ускоренный процесс рассмотрения национальных IDN-доменов верхнего уровня (IDN ccTLD Fast Track Process).

Многие правила и политики, разработанные и утвержденные вышеперечисленными группами и комитетами ICANN, влияют на содержимое корневой зоны. Они определяют процессы внесения изменений, а также осуществление обслуживания корневой зоны DNS — об этом мы поговорим в следующих разделах.

Корневой уровень DNS

Корневая зона содержит информацию обо всех доменах верхнего уровня: национальных доменах (например, .ru, .рф), доменах общего назначения (например, .com, .museum, .москва, .дети[72]). Точнее, эта информация содержит списки серверов имен, обслуживающих тот или иной домен верхнего уровня. Таким образом клиенту указывается, на какие серверы DNS отправить последующий запрос для продолжения разрешения полного доменного имени. Как мы видели, говоря о процессе разрешения имени, любой «свежий» (то есть не сохраненный в кеше резолвера) запрос начинается с обращения к так называемым корневым DNS-серверам (КС), обеспечивающим доступ к зоне.

Мы не напрасно отметили, что запрос является «свежим». Дело в том, что обычно резолвер запоминает ответы, полученные от серверов DNS, и на по-

[70] https://www.icann.org/consensus-policies-ru

[71] https://www.iana.org/help/cctld-delegation

[72] Сегодня домены общего назначения (generic top-level domains, gTLD) включают исторически общие домены, такие как .com и .net, спонсированные домены (.aero, .coop), географические домены (.cat, .asia), а также домены, созданные в рамках начатой в 2008 г. программы ICANN по масштабному созданию доменов верхнего уровня.

вторные запросы отвечает данными из своего кеша. Время хранения ответов определяется администратором соответствующего домена и в случае корневой зоны для большинства записей равняется 48 часам.

Система корневых серверов и координация ее работы

Генерация и распределение корневой зоны

Состав корневой зоны постоянно меняется. В среднем в зону вносится несколько изменений в неделю. Например, изменяется информация о DNS-серверах, обслуживающих домен верхнего уровня. Или же требуется добавление нового домена верхнего уровня, хотя такие изменения происходят гораздо реже. Как это происходит?

Запрос на изменение поступает от администратора домена верхнего уровня (ccTLD, gTLD и т.д.) и обслуживается IANA (Internet Assigned Numbers Authority) — структурой, отвечающей за регистрацию изменений в корневой зоне, оператором которой является ICANN[73].

После проведения необходимых административных и технических процедур (например, проверки правильности и законности запроса, проверки возможных негативных последствий для корневой зоны) запрос на изменение подписывается цифровым образом и направляется в организацию, ответственную за публикацию зоны в DNS, называемой Root Zone Maintainer (организация по обслуживанию корневой зоны). Эту роль в настоящее время выполняет компания VeriSign по контракту с ICANN[74].

Зона публикуется на скрытом мастер-сервере и затем распространяется на все корневые серверы с использованием протокола TSIG, защищающего данные от модификации при передаче. Независимо от наличия или отсутствия изменений корневая зона обновляется дважды в день.

Этот процесс схематично представлен на рис. 30.

---

[73] В марте 2014 г. NTIA объявило о намерении передать ряд функций, связанных с управлением IANA, глобальному сообществу (см. http://ntia.doc.gov/press-release/2014/ntia-announces-intent-transition-key-internet-domain-name-functions). ICANN было поручено начать диалог со всеми заинтересованными группами для поиска модели системы управления IANA без участия NTIA. В результате этого процесса ICANN была создана дочерняя организация PTI (Public Technical Identifiers), выполняющая функции IANA. Об этом более подробно рассказано в разделе «Передача ключевых функций» в главе 4.

[74] https://www.icann.org/iana_imp_docs/129-root-zone-maintainer-service-agreement-v-28sep16

Рис. 30. Процесс внесения изменений в корневую зону.
Источник: Отчет об исследовании процесса внесения изменений в корневую зону (Root Zone Update Process Study, https://itp.cdn.icann.org/en/files/internet-assigned-numbers-authority-iana-functions/rzm-study-jas-icj-14-03-2022-14-03-2022-en.pdf)

Корневые серверы (КС)

Корневую зону обслуживают 13 DNS-серверов, также называемых корневыми. Имена КС начинаются с буквы латинского алфавита (от A до M) и имеют общее окончание root-servers.net, например, a.root-servers.net. Эта первая буква также используется как сокращенное имя сервера, например, сервер k.root-servers.net называют «K-сервер». Операторами КС являются различные организации, получившие право управления серверами в относительно отдаленном прошлом, когда подобные вопросы решались менее формально. Среди операторов — университеты, организации Минобороны США, некоммерческие ассоциации. За исключением оператора L-сервера, который находится под управлением ICANN, операторы финансово и юридически независимы от корпорации ICANN, в рамках которой действует IANA. При принятии операционных решений операторы руководствуются технической целесообразностью и существующими стандартами (например, RFC 2870[75]), в основном поддерживая статус-кво. Принято считать, что подобная независимость и разнородность операторов КС является основой технической и политической стабильности системы в целом, исключая узурпацию управления какой-либо из сторон.

Операторы КС образуют неформальную группу, цель которой — координация совместных действий и обмен операционной информацией и опытом. Группа

[75] RFC 2870: Root Name Server Operational Requirements, URL: https://www.rfc-editor.org/rfc/rfc2870

регулярно, три раза в год, проводит встречи, приуроченные к совещанию IETF. Одним из результатов таких совещаний является генерация секретного ключа KSK (Key Signing Key) для протокола TSIG. Члены группы также входят в Консультационный совет KSK ICANN (Root Server System Advisory Committee, RSSAC), среди задач которого — выработка рекомендаций по управлению KSK и внесению различных изменений в систему.

До недавнего времени отсутствовали какие-либо формальные отношения между операторами и ICANN/IANA. Эта ситуация изменилась с подписанием первого соглашения между ICANN и оператором F-сервера компанией ISC[76]. Данное соглашение не предусматривает никаких финансовых расчетов и лишь определяет взаимные обязанности сторон в отношении управления КС. Ряд операторов также обменялись письмами о взаимопонимании с ICANN (см., например, письмо от RIPE NCC[77]).

Ниже приведен список и краткая характеристика текущих операторов SKS (Synchronising Key Server – синхронизирующий сервер ключей)

Таблица 5. Список операторов SKS

КС	Организация-оператор	Характер деятельности, страна
A	VeriSign, Inc.	Коммерческая корпорация, один из крупнейших операторов DNS (например, .com, .net), поставщик средств защиты электронных коммуникаций, США
B	Information Sciences Institute	Институт Университета Южной Калифорнии (USC), США
C	Cogent Communications	Один из крупнейших коммерческих интернет-сервис-провайдеров, США
D	University of Maryland	Мэрилендский университет, США
E	NASA's Ames Research Center	Государственное агентство, США
F	Internet Systems Consortium, Inc.	Некоммерческая корпорация, США
G	U.S. DOD Network Information Center	Государственное агентство, США
H	U.S. Army Research Lab	Государственное учреждение, США
I	Netnod	Коммерческая организация, оператор точки обмена трафиком, оператор DNS, Швеция
J	VeriSign, Inc.	Коммерческая корпорация, один из крупнейших операторов DNS (например, .com, .net), поставщик средств защиты электронных коммуникаций, США
K	RIPE NCC	Некоммерческая ассоциация, РИР, Нидерланды
L	ICANN	Некоммерческая корпорация, США
M	WIDE Project	Некоммерческий проект, секретариат Университета Кейо, Япония

[76] https://www.icann.org/en/announcements/details/icann-board-approves-historic-f-root-agreement—first-formalization-of-mutual-responsibilities-between-isc-f-root-server-operator-and-icann-23-1-2008-en

[77] https://www.icann.org/en/system/files/files/pawlik-to-twomey-06may09-en.pdf

Альтернативные SKS

Так называемые альтернативные SKS появились из-за неудовлетворенности существующей системой управления SKS во главе с ICANN при участии правительства одной страны — США, а также по причине географической распределенности серверов и недостаточной поддержки интернационализированных доменов. Некоторые из таких SKS существуют до сих пор, например, Public-Root или Open Root Server Network (ORSN). Хотя эти системы копируют текущее состояние корневой зоны, сама архитектура предусматривает, что в определенных условиях альтернативные SKS могут предоставить альтернативное пространство имен. Администратор клиента DNS (обычно — сервера DNS, обслуживающего корпоративных пользователей или клиентов кабельной сети) может выбрать альтернативную SKS, просто изменив соответствующим образом файл hints.

Альтернативные SKS получили критическую оценку со стороны IETF как открывающие потенциальную возможность раскола единого Интернета (см. RFC 2826[78]).

Надо заметить, что масштабное внедрение аникаста в системе корневых серверов, а также поддержка ICANN интернационализированных имен существенно уменьшили необходимость в альтернативных серверах. Тем не менее, различные политические пертурбации, например, разоблачения Эдварда Сноудена, время от времени повышают активность групп, связанных с альтернативными SKS.

Экспериментальная СКС — Yeti DNS

Хотя за более чем 25-летнее существование в SKS произошел ряд существенных изменений — внедрение технологии аникаста, поддержка IPv6, подписание корневой зоны DNSSEC, — все они несли в себе риски, несмотря на значительную предварительную подготовку и тестирование. Причиной этому является то, что система SKS — единственная в своем роде, и лабораторное моделирование не способно отразить все многообразие экосистемы, в которой она существует. Взять хотя бы разнообразие DNS-клиентов или непредсказуемость различных промежуточных устройств — защитных сетевых экранов, балансировщиков нагрузки и т.п. Вследствие этого точно определить последствия того или иного изменения в системе SKS невозможно. Поскольку стабильность системы является наиболее важным требованием, возможность экспериментирования в SKS сводится к нулю. Это, в свою очередь, ведет к оссификации системы и затруднению ее дальнейшего развития.

В 2015 году несколько организаций (WIDE, BII и TISF) начали строительство экспериментальной SKS, получившей название Yeti DNS[79]. Эта система ис-

[78] RFC 2826: IAB Technical Comment on the Unique DNS Root, URL: https://www.rfc-editor.org/rfc/rfc2826

[79] https://yeti-dns.org

пользует точную копию официальной корневой зоны IANA (с точностью до замены адресов корневых серверов). Хотя по своей структуре она напоминает альтернативные SKS, задачей Yeti DNS является создание вовсе не альтернативного пространства имен, а параллельной системы для проведения экспериментов.

В частности, Yeti DNS может помочь ответить на следующие вопросы:

- Можно ли обеспечить работу SKS, используя исключительно IPv6?
- Каковы последствия более частой замены DNSSEC-ключей ZSK, например, каждые две недели?
- Каковы последствия более частой замены DNSSEC-ключей KSK, например, каждые шесть недель?
- Каково оптимальное число корневых серверов?
- Каковы последствия добавления или удаления оператора корневого сервера, насколько часто это можно делать?

Несколько опытов были успешно проведены, но проект испытывает ряд трудностей. Это связано, в первую очередь, с недостаточным объемом запросов, чтобы имитировать работу реальной SKS. Проблема усугубляется тем, что некоторые эксперименты могут отрицательно отразиться на качестве предоставляемой услуги. В результате некоторые участники Yeti DNS перестают использовать эту систему и переключаются на официальную SKS. Это, в свою очередь, еще более уменьшает нагрузку на Yeti DNS. В настоящее время проект продолжается, хотя и менее активно.

Локальный корень

Как мы уже видели, корневые серверы играют ключевую роль в процессе трансляции имен. Для каждого запроса к доменному имени для домена верхнего уровня, отсутствующего в кеше резолвера, этот запрос сначала отправляется к корневому серверу. Хотя время жизни ответов, полученных от корневых серверов, варьируется от одного до двух дней, отсутствие доступа к корневым серверам, например, вследствие атаки DDoS, приведет к отказу в обслуживании для всех доменов, чьи TLD отсутствуют в кеше резолвера. А если атака продлится достаточно долго, эта участь постепенно постигнет все запросы.

С другой стороны, подавляющая часть запросов к корневым серверам относится к несуществующим доменам, как следствие, значительные ресурсы системы тратятся впустую.

Документ «Запуск локального корневого сервера»[80] описывает подход, позволяющий решить обе проблемы.

[80] RFC 8806: Running a Root Server Local to a Resolver, URL: https://www.rfc-editor.org/rfc/rfc8806

Суть подхода заключается в том, что оператор рекурсивного резолвера имеет полную корневую зону локально, тем самым исключая необходимость внешних запросов к SKS. Основная идея состоит в том, чтобы создать службу постоянно обновляемой корневой зоны на том же хосте, что и резолвер, и использовать эту службу, когда резолверу требуется информация корневой зоны. Резолвер проверяет все ответы от корневой службы на том же хосте точно так же, как он проверяет все ответы от удаленного корневого сервера.

При этом необходимо учитывать несколько аспектов:

Во-первых, эта служба корневой зоны должна быть сконфигурирована таким образом, чтобы предотвратить доступ к ней какой-либо другой системы, кроме резолвера на том же хосте.

Во-вторых, содержимое корневой зоны должно обновляться с использованием таймеров из записи SOA в корневой зоне. По сути, это означает, что содержимое локальной корневой зоны, скорее всего, будет немного отставать от содержимого глобальных корневых серверов, поскольку эти серверы обновляются динамично, используя сообщения NOTIFY.

Корневую зону можно получить откуда угодно, если она содержит все записи DNSSEC, необходимые для проверки. В настоящее время можно получить корневую зону от ICANN путем передачи зоны AXFR[81] по протоколу TCP с DNS-серверов по адресам xfr.lax.dns.icann.org и xfr.cjr.dns.icann.org. Файл корневой зоны можно получить с помощью методов, описанных на странице Root Files[82]. В настоящее время корневая зона также может быть получена с помощью AXFR по TCP от следующих корневых серверов:

- b.root-servers.net
- c.root-servers.net
- d.root-servers.net
- f.root-servers.net
- g.root-servers.net
- k.root-servers.net

Здесь мы подходим к еще одной проблеме – каким образом удостовериться в целостности полученной локальной зоны. Простым ответом кажется использование DNSSEC. Однако не все записи зоны подписаны DNSSEC. Так, записи NS делегированных доменов и их IP-адреса не подписываются.

Другие способы защищенной передачи зоны, такие как TSIG, эфемерны и гарантируют лишь то, что клиент получит данные от ожидаемого сервера и что

[81] RFC 5936: DNS Zone Transfer Protocol (AXFR), URL: https://www.rfc-editor.org/rfc/rfc5936

[82] https://www.iana.org/domains/root/files

данные, отправленные сервером, не будут изменены во время передачи. Однако они не гарантируют, что сервер передает данные в первоначально опубликованном виде, и не предоставляют никаких методов для проверки данных, которые используются после завершения передачи.

Здесь на помощь приходит новая запись DNS – ZONEMD, – описанная в стандарте «Дайджест сообщений для зон DNS»[83]. Запись ZONEMD является криптографическим дайджестом данных в зоне. Это позволяет получателю зоны проверить целостность и подлинность зоны при использовании в сочетании с DNSSEC. ZONEMD является частью самой зоны, что позволяет проверять зону, независимо от того, каким способом она была получена.

Подписание корневой зоны

Как мы уже обсуждали, существенным недостатком базового протокола DNS является слабая система защиты данных. Ответы могут быть модифицированы «в полете» или путем создания ложных серверов. Их проникновение в кеши резолверов (так называемое отравление кеша) может производить продолжительный эффект. Радикальным решением проблемы является использование технологии DNSSEC, которая позволяет получателю ответа определить, были ли данные модифицированы.

В силу иерархического характера DNS подписание записей корневой зоны имело существенное значение как для усиления безопасности глобальной системы DNS, так и для более широкого внедрения DNSSEC.

Многие годы велись дискуссии о том, когда же будет подписана зона, кто будет контролировать ключи, как это отразится на системе DNS и Интернете в целом. После долгих публичных комментариев и внутренних дискуссий в 2009 году ICANN, VeriSign и Министерство торговли США договорились о прагматичной схеме, при которой существующий процесс внесения изменений в зону остается прежним, а основные игроки получают дополнительные роли: ICANN контролирует так называемый Trust Anchor — ключ для подписания ключей KSK, агентство Министерства торговли NTIA по-прежнему утверждает изменения, а VeriSign владеет ключом подписания зоны ZSK, который используется для генерирования подписанной корневой зоны, и осуществляет ее публикацию на скрытом мастер-сервере. Далее зона публикуется операторами корневой зоны. Надо отметить, что в соответствии с технологией DNSSEC ключ KSK подписывает ключ(и) ZSK. Другими словами, контроль за подписанием зоны в конечном итоге остается за ICANN.

Когда политические страсти вокруг подписания корневой зоны постепенно улеглись, настало время взглянуть на технические аспекты этого изменения. А их немало. Помимо внутренней защищенной архитектуры хранения и ис-

[83] RFC 8976: Message Digest for DNS Zones,
URL: https://www.rfc-editor.org/rfc/rfc8976

пользования ключей, а также защищенного взаимодействия между игроками, сама публикация подписанной зоны представляет серьезную задачу. Это, в первую очередь, связано с масштабом последствий, которые изменения в корневой зоне могут иметь для глобального сообщества пользователей Интернета. Специально созданная техническая группа, состоящая из экспертов в области DNS и безопасности, провела колоссальную работу по подготовке и воплощению данного масштабного проекта в жизнь.

Одной из основных задач, стоявших перед группой, являлось обеспечение постепенного внедрения технологии DNSSEC в корневой зоне. Было очевидно, что простая публикация подписанной зоны в один прекрасный момент была бы слишком рискованной и поэтому неприемлемой. Как поведут себя при этом разнообразные клиенты? Как это скажется на корневых серверах?

Как ни странно, основным изменением, связанным с внедрением DNSSEC в корневой зоне, являлось не собственно подписание зоны, а увеличение размера ответа на запрос клиента. Большие DNS-ответы подстерегают различные опасности: это и фрагментация пакетов на пути их следования, и невозможность их сборки клиентом, и фильтрация пакетов, превышающих по длине исторические 512 байт, маршрутизаторами и устройствами безопасности. Другими словами, при значительном увеличении размера ответов возрастает риск, что клиент не сможет получить ответ на запрос к корневому серверу.

Поэтому DNSSEC в корневой зоне было решено внедрять постепенно: сначала на одном сервере, потом на следующем и так далее, пока подписанная зона не будет публиковаться всеми 13 корневыми серверами (точнее, внедрение происходило по группам — всего шесть групп серверов). При этом велись наблюдения за возможным перераспределением нагрузки на серверы, чтобы исключить возможные осложнения. Ведь структурная «эмиграция» клиентов от сервера, на котором опубликована подписанная зона, скорее всего, означает, что клиенты либо не могут получить ответа от этого сервера, либо выполнение запроса занимает существенно больше времени, чем для остальных серверов.

Также следовало обеспечить возможность возвращения «на круги своя» — то есть к неподписанной корневой зоне — в случае каких-либо проблем. Было принято решение подписать зону таким образом, чтобы подписи не могли никоим образом подвергнуться проверке. Такую зону назвали DURZ — Deliberately Unvalidatable Root Zone, или корневая зона, которая не может быть криптографически проверена.

Процесс «распространения» DURZ по КС начался в январе 2010 года и занял почти полгода. 5 мая 2010 года DURZ была опубликована на последнем сервере — «J». Все прошло без каких-либо заметных негативных последствий в функционировании Интернета. Можно было переходить к следующему этапу — подписанию зоны «правильным» ключом.

16 июня семь доверенных представителей интернет-сообщества прибыли на площадку ICANN в местечке Калпепер (Culpeper) недалеко от Вашингтона для церемонии генерации ключей KSK и подписания ключей ZSK, которые в свою очередь используются для подписания записей самой корневой зоны.

15 июля 2010 года корневая зона была подписана этим ключом. Следуя передовой криптографической практике, ключи подписания зоны должны периодически обновляться. Новый пакет ключей ZSK генерируется четыре раза в год группой доверенных представителей интернет-сообщества. К концу 2016 года были проведены 27 таких церемоний, включая самую первую, в ходе которой был также сгенерирован первый ключ KSK. Однако и этот ключ необходимо периодически, хотя и гораздо реже, менять.

Однако замена ключа KSK является нетривиальной задачей. Дело в том, что в то время, как новые ключи ZSK (как и другие записи ресурсов) обновляются клиентом в ходе нормальных запросов DNS, ключ KSK, который является точкой доверия всей иерархии глобальной системы имен, конфигурируется администратором каждого резолвера вручную. До публикации стандарта RFC 5011[84] не существовало способа автоматического обновления KSK. Можно предположить, что не все резолверы поддерживают этот протокол, что делает задачу распределения нового KSK дополнительно сложной.

Для решения этой задачи группой экспертов был разработан план замены KSK, который был передан на обсуждение интернет-сообщества. План включал следующие шаги:

- Октябрь 2016: начало процесса замены KSK: генерация нового ключа KSK.
- Июль 2017: публикация нового KSK в DNS.
- Октябрь 2017: новый KSK используется для подписания пакета ключей корневой зоны, что, собственно, и означает замену ключа.
- Январь 2018: отзыв старого KSK.
- Март 2018: завершение процесса замены KSK.

Точка замены ключа (октябрь 2017 года в изначальном плане) является наиболее критическим моментом. Резолверы, осуществляющие проверку DNSSEC и не сконфигурировавшие новый ключ KSK (автоматически, используя протокол RC5011, или вручную администратором), выдадут ошибку валидации на любой запрос.

Эта проблема явилась причиной того, что замену ключа пришлось отсрочить на один год. Дело в том, что в начале 2017 года в IETF был разработан механизм, позволяющий резолверам, осуществляющим проверки DNSSEC, проинформи-

[84] RFC 5011: Automated Updates of DNS Security (DNSSEC) Trust Anchors, URL: https://www.rfc-editor.org/rfc/rfc5011

ровать DNS-сервер о том, какие ключи они используют для валидации[85]. Используя этот механизм, исследователи смогли получить более точное представление о проценте резолверов, готовых к замене ключа. К сожалению, как видно из рис. 31, на предполагаемый момент замены значительная часть резолверов (6-8%, это число выросло до 20% по мере увеличения числа измерений) по-прежнему доверяла старому ключу, а не новому. Этот процент начал уменьшаться только в мае 2018 года. На момент новой даты замены ключа – 11 октября 2018 года – только 5% наблюдаемых резолверов по-прежнему доверяли старому ключу, что являлось приемлемым для процесса замены.

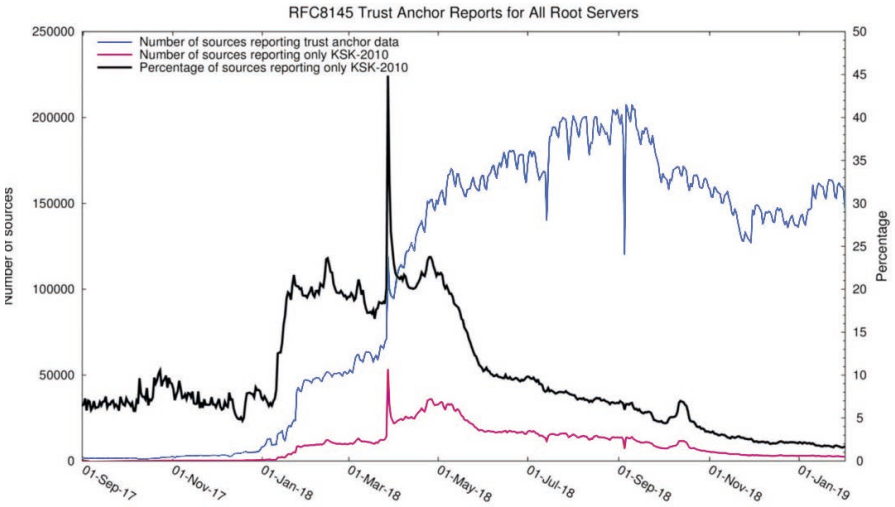

Рис. 31. Отчеты резолверов по использованию ключей KSK.
Источник: Обзор замены ключа DNSSEC KSK 2018 г.
https://www.icann.org/review-2018-dnssec-ksk-rollover.pdf

11 октября 2018 года произошла замена ключа. Ключи, используемые для подписания записей зоны, были подписаны новым ключом KSK. Этот момент также означал, что резолверы, которые не завершили переход, стали возвращать ошибку при попытке валидации DNSSEC. Тем не менее, процесс прошел без существенных инцидентов. По данным проекта Root Canary[86], в течение 48 часов после замены ключа 99% наблюдаемых резолверов использовали новый ключ[87].

[85] RFC 8145: Signaling Trust Anchor Knowledge in DNS Security Extensions (DNSSEC), URL: https://www.rfc-editor.org/rfc/rfc8145

[86] Проект был завершен в начале 2019 года, https://rootcanary.org/

[87] https://www.sidnlabs.nl/en/news-and-blogs/a-successful-root-ksk-rollover-a-short-look-back??language_id=2

Итак, замена ключа KSK прошла успешно и в будущем превратится в более или менее рутинную операцию. Гораздо более сложной является операция по замене криптографического алгоритма ключа. При первой замене ключа по-прежнему использовался широко распространенный алгоритм криптографической подписи RSA-SHA. В последнее время более распространенными стали новые криптографические алгоритмы. Это привело к обсуждению в сообществе ICANN необходимости подготовки процесса обновления будущих криптографических ключей DNS для реализации преимуществ этих новых алгоритмов.

В ноябре 2022 года ICANN пригласила добровольцев присоединиться к группе разработчиков плана изменения криптографического алгоритма, используемого для ключа подписи корневого ключа DNS и ключа подписи зоны. В конце февраля 2023 года такая группа была сформирована, о чем ICANN сообщила в своем анонсе[88].

Глобализация корневой зоны DNS. Программа gTLD

В 1984 году документ RFC 920 «Требования к доменам»[89] определил дополнительные домены верхнего уровня. К существовавшему домену .arpa, включавшему все хосты тогдашнего Интернета, были добавлены пять «смысловых» доменов: .gov для правительственных ресурсов, .edu для образовательных, .mil для военных, .com для коммерческих и .org для организаций. Разумеется, тогда все это относилось к американским организациям.

Данный документ также определил национальные домены и установил их формат — двухбуквенный код (alpha-2) таблицы ISO-3166.

Наконец, в документе была упомянута тогда еще пустая категория «мультиорганизаций» — больших транснациональных конгломератов, «не попадающих ни в одну из перечисленных категорий». До середины 1990-х годов корневая зона росла за счет национальных доменов, за исключением добавленных доменов .int и .net. Как заявил Джон Постел в 1994-м, «крайне маловероятно, что будут созданы новые домены верхнего уровня»[90]. Однако требования к созданию дополнительных доменов верхнего уровня значительно выросли, что привело к созданию нескольких групп и разработке соответствующих предложений.

Основной причиной этих требований явилась неудовлетворительная, с точки зрения мировой общественности, ситуация с так называемыми международными доменами. Именно таковыми со временем стали домены .com, .org и .net. Интернет переживал бум приватизации и стремительного развития, поэтому

[88] https://www.icann.org/ru/announcements/details/icann-convenes-design-team-to-evolve-root-zone-security-21-02-2023-ru

[89] RFC 920: Domain Requirements, URL: https://www.rfc-editor.org/rfc/rfc920

[90] RFC 1591: Domain Name System Structure and Delegation, URL: https://www.rfc-editor.org/rfc/rfc1591

регистрация имен и «международных» поддоменов стала носить глобальный характер. Но регистрация осуществлялась единственной центральной регистратурой Internic, обслуживаемой частной компанией Network Solutions (сегодня VeriSign). Проблема монополизации «международных» доменов требовала решения. Регулирование было невозможно, учитывая международный характер проблемы. Следовало привлекать рыночные механизмы. Решению вопроса «открытия» рынка корневой зоны и были посвящены новые предложения.

Одним из таких предложений явился так называемый Проект Постела[91], разработанный Джоном Постелом в 1996 году. Проект предусматривал создание нескольких комитетов, утверждающих образование новых доменов верхнего уровня. Решение носило технократический характер и предлагало осуществлять делегирование новых доменов верхнего уровня в том же стиле, в каком IANA назначает параметры протоколов. Проект Постела также предлагал передать управление IANA созданной в 1992 году организации ISOC (Internet Society) — «организационному дому» IETF. Этот проект вызвал критику со стороны общественности как ограничивающий конкуренцию.

Другое предложение было разработано группой под названием International Ad Hoc Committee (IAHC), специально созданной под эгидой ISOC, IAB, IANA, ITU, INTA и WIPO. Здесь была сделана попытка учесть недостатки проекта Постела и предложить более сбалансированную модель управления. Кстати, именно IAHC предложил использование термина «общие домены верхнего уровня» (gTLD, generic TLD) вместо «международных» (iTLD, international TLD), использовавшихся ранее. Комитет прекратил свое существование в мае 1997 года после публикации предложения. Хотя рекомендации этой группы не были воплощены в жизнь, многие из них легли в основу деятельности созданной вскоре корпорации ICANN[92].

Прошло чуть меньше двух лет после создания ICANN, и уже в 2000 году было анонсировано создание семи новых имен: .aero, .biz, .coop, .info, .museum, .name, .pro, которые постепенно появились в корневой зоне к 2003 году.

Следующий этап расширения корневой зоны прошел под флагом так называемых спонсированных доменов. Спонсорами этих имен являлись этнические, профессиональные или географические сообщества.

Наконец, в 2005 году gNSO начала рассмотрение вопроса о более масштабном создании общих доменов верхнего уровня. Эта работа базировалась на опыте внедрения ICANN доменов в 2000–2003 годах, а процесс разработки соответствующих политик занял два года. В результате были утверждены 19 рекомен-

[91] https://datatracker.ietf.org/doc/draft-postel-iana-itld-admin/

[92] С текстом предложения можно ознакомиться на https://web.archive.org/web/20070426031850/http://www.gtld-mou.org:80/draft-iahc-gTLDspec-00.html

даций по созданию новых gTLD, включающих критерии выбора имени и договорные условия. Программа создания новых gTLD вызвала немало критики со стороны общественности. Практические ответы на многие вопросы — о защите прав правообладателей, финансовых последствиях регистрации тех или иных имен в качестве доменов верхнего уровня и т.п. — нам еще предстоит увидеть. Программа также вызвала обеспокоенность технического сообщества. Для ответа на наиболее острые вопросы, связанные с устойчивостью и масштабируемостью корневой зоны и SKS, было предпринято несколько исследований как самой корпорацией ICANN, так и комитетами внутри корпорации — SSAC и RSSAC[93]. Результаты этой работы, разумеется, содержат элемент неопределенности. Но стало очевидным, что расширение корневой зоны примерно на 1000 новых доменов в год хотя и требует периодической переоценки и мониторинга, на данном этапе не представляет существенного технического риска.

В июне 2011 года совет директоров ICANN объявил запуск программы новых gTLD. Прием заявок на них был начат 12 января 2012 года. В итоге было получено 1930 заявок. На июнь 2014 года 272 заявки были реализованы в виде новых доменов gTLD, таких как .website, .vodka, .club, .luxury, .москва, .дети и т.д. Спустя два года число доменов верхнего уровня выросло более чем в четыре раза, достигнув почти 1200 доменов верхнего уровня. Однако после первоначального ажиотажа рост существенно замедлился, и с тех пор добавилось только несколько десятков доменов.

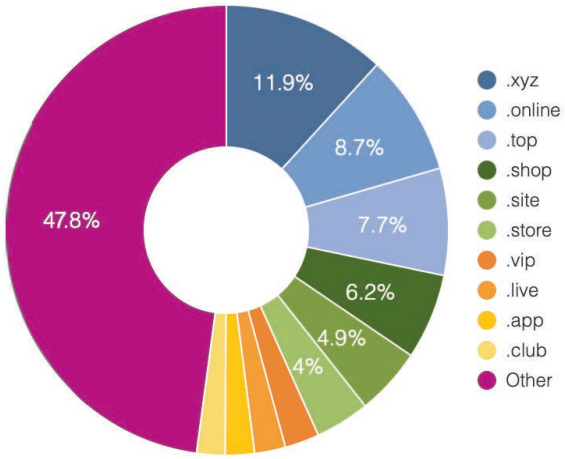

Рис. 32. Доля регистраций доменов второго уровня в new gTLD (июль 2023).
Источник: https://ntldstats.com/tld

93 https://www.icann.org/en/announcements/details/new-gtlds-root-zone-scaling-report-27-6-2012-en

Очевидно, что емкость рынка ограничена, и больше половины всех доменов второго уровня зарегистрированы в десяти крупнейших gTLD.

Однако на повестке дня стоит подготовка к следующему раунду программы new gTLD. Этому предшествовала значительная работа по подведению итогов раунда 2012 года и выработка рекомендаций по улучшению.

Уже в конце 2015 года Совет gNSO инициировал процесс разработки политики и создал рабочую группу по последующим процедурам для новых gTLD. Перед рабочей группой была поставлена задача использовать коллективный опыт сообщества, полученный в ходе первого раунда программы New gTLD 2012 года, чтобы определить, какие изменения, если таковые имеются, необходимо внести в существующие рекомендации по политике введения новых родовых доменов верхнего уровня от 8 августа 2007. Эта работа была завершена 18 февраля 2021 года утверждением и публикацией «Итогового отчета о процессе разработки политики последующих процедур для программы new gTLD»[94].

Следующим шагом является работа по реализации рекомендаций, содержащихся в итоговом отчете, в процессе разработки политики последующих процедур для новых gTLD. Результатом этого процесса станет обновленное «Руководство кандидата» (Applicant Guidebook, AGB). Также ICANN работает над планом реализации нового раунда.

Учитывая все эти задачи, корпорация ICANN ожидает, что AGB будет завершено во втором квартале 2025 года, что позволит начать прием заявок во втором квартале 2026.

Заключение

Глобальная система доменных имен DNS имеет интересное свойство, противоположное сетевой самоорганизующейся архитектуре Интернета. DNS и в архитектурном и, что более важно, в операционном смыслах — иерархическая, хотя и распределенная система. В DNS существует только один корень, и проблемы на этом уровне затрагивают всю систему.

Тем не менее, эта система успешно идет в ногу с самим Интернетом без фундаментальных изменений во внутренних протоколах и архитектуре. Более того, в качестве глобальной распределенной базы данных DNS уже давно не ограничивается исполнением той простой функции, для которой

[94] Final Report on the new gTLD Subsequent Procedures Policy Development Process, https://gnso.icann.org/sites/default/files/file/field-file-attach/final-report-newgtld-subsequent-procedures-pdp-20jan21-en.pdf

была создана, — трансляции имен в адреса IP. Система DNS активно используется во многих новациях: как в хороших — например, при оптимизации доставки контента, так и в плохих — например, в целях управления ботнетами в руках злоумышленников.

Еще одной особенностью DNS является то, что эта система всегда на виду — она наиболее понятна пользователю Интернета. Мы набираем доменное имя и получаем доступ к веб-серверу или другому ресурсу. Не случайно вопросы интернационализации DNS довольно остро стояли и отчасти продолжают стоять на повестке дня. Проблематика DNS также широко обсуждается и на политическом уровне — в рамках дискуссий об управлении Интернетом.

Другие аспекты DNS, возможно, менее заметны, но успешное решение связанных с ними вопросов не менее важно. В первую очередь имеются в виду вопросы безопасности. Хотя спецификации расширений безопасности DNS — DNSSEC — были стандартизованы IETF еще в 2005 году, уровень их внедрения не внушает большого оптимизма. А ведь DNSSEC может открыть новые возможности использования DNS — в частности, с применением сертификатов TLS веб- и других ресурсов в соответствии со спецификациями DANE.

И пусть иногда кажется, что поисковые машины постепенно уменьшают значимость DNS, заменяя адреса ресурсов Сети на ключевые слова — объекты запросов, но в реальности DNS вряд ли скоро уступит свои позиции.

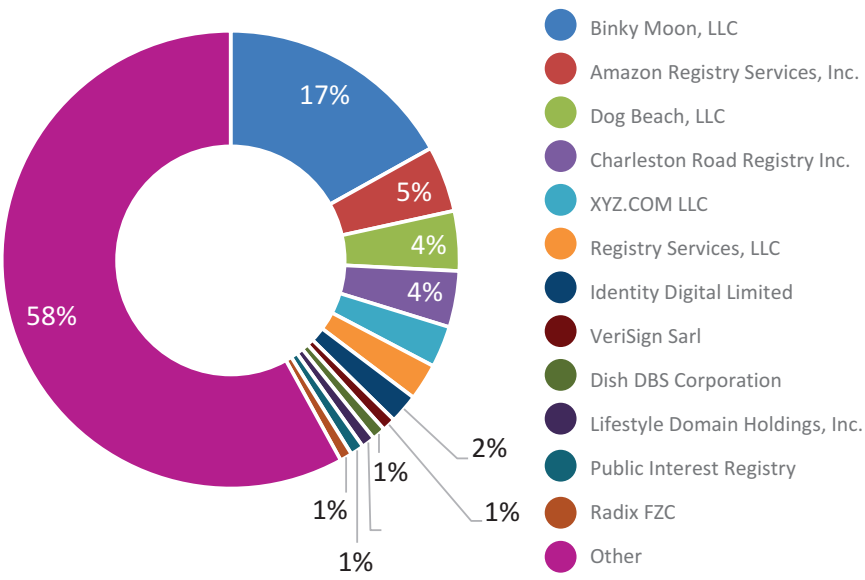

Рис. 33. Распределение рынка доменов new gTLD между операторами регистратур (состояние март 2023 г.).
Источник: www.icann.org/en/system/files/files/rr-voting-status-xls-20mar23-en.xlsx

Очевидно, что емкость рынка ограничена, и больше половины всех доменов второго уровня зарегистрированы в десяти крупнейших gTLD, как видно из рисунка 32.

И в целом рынок new gTLD является весьма сконцентрированным. Дюжина операторов регистратур владеют 42% всех доменов new gTLD, см. рис 33.

Подобная ситуация наблюдается и на рынке регистраторов. Почти 60% рынка всех зарегистрированных доменов второго уровня принадлежит 10 компаниям (рис 34).

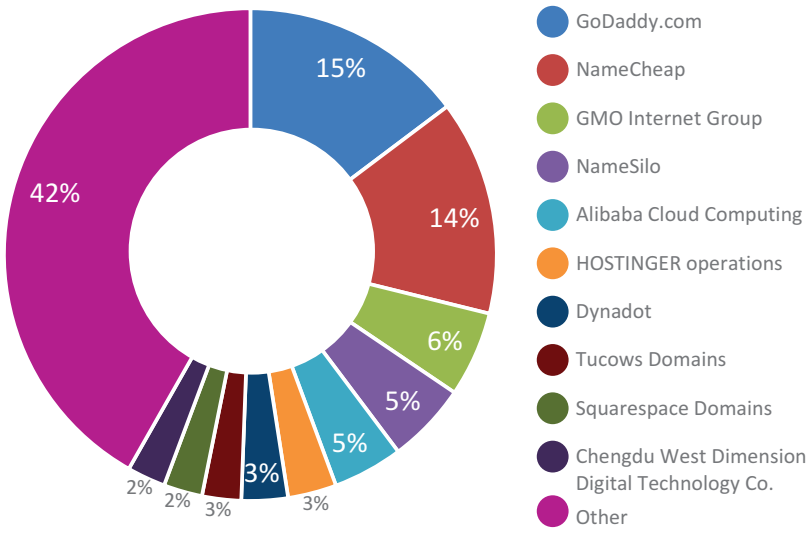

Рис. 34. Распределение рынка new gTLD между регистраторами по числу доменов второго уровня (состояние февраль 2024 г.).
Источник: https://ntldstats.com/registra


ПРОВАЙДЕР
IRR
АНОНС
ПИРИНГ
АНОНС
BGPSEC
СВЯЗНОСТЬ
ОБМЕН ТРАФИКОМ
ФИЛЬТР
СЕРТИФИКАТ РЕСУРСА
ТЕХНИЧЕСКАЯ ПЛОЩАДКА
ЗАТОР
IRR
ТРАНЗИТ
ПИРИНГ
BGP
MSK-IX
АТРИБУТ
УЗЕЛ
МАРШРУТ
АВТОНОМНАЯ СИСТЕМА
ОБЛАКА
КЛЮЧ
OPENFLOW
ТРАНЗИТ
ЗАХВАТ ПРЕФИКСА
РАСПРЕДЕЛЕНИЕ КОНТЕНТА
ПРЕФИКС
ОБЛАКА
ARIN
УЗЕЛ
DIFFSERV
АВТОНОМНАЯ СИСТЕМА
МАРШРУТИЗАТОР
СВЯЗНОСТЬ
МАРШРУТ
CODEL
DIFFSERV
INTSERV
BGPSEC
МАРШРУТИЗАЦИЯ
ПРЕФИКС
BGPSEC
АНОНС
УЗЕЛ
АТАКА
AS_PATH
ЗАТОР
КЛЮЧ
CODEL
NSFNET
АВТОНОМНАЯ СИСТЕМА
ROUTE SERVER
РАСПРЕДЕЛЕНИЕ КОНТЕНТА
OPENFLOW
РАСПРЕДЕЛЕНИЕ КОНТЕНТА
ПРОПУСКНАЯ СПОСОБНОСТЬ
СЕРТИФИКАТ РЕСУРСА
МАРШРУТИЗАТОР
СЕРТИФИКАТ РЕСУРСА
СВЯЗНОСТЬ
ПИР
КАНАЛ
ROUTE SERVER
СЕРТИФИКАТ РЕСУРСА
ПИРИНГ
КЛИЕНТ
УТЕЧКА МАРШРУТА
ИНФРАСТРУКТУРА
ЗАХВАТ ПРЕФИКСА
NSFNET
СЕРТИФИКАТ РЕСУРСА
АВТОНОМНАЯ СИСТЕМА
МАРШРУТИЗАТОР
ПРОПУСКНАЯ СПОСОБНОСТЬ
IRR
OPENFLOW
ROUTE SERVER
ПРОВАЙДЕР
OPENFLOW
DIFFSERV
BGPSEC
BGPSEC
РАСПРЕДЕЛЕНИЕ КОНТЕНТА

Глава 3

Глобальная система маршрутизации и передачи данных

> В недалеком будущем мы будем рассматривать Интернет как набор автономных систем.
>
> *RFC 827[1], октябрь 1982 г.*

Принципы маршрутизации данных в Интернете

Когда говорят о маршрутизации данных, обычно имеют в виду процесс определения маршрута пакетов — от источника к получателю. В Интернете этот маршрут состоит из множества участков: каждый из маршрутизаторов по пути предполагаемого следования данных определяет его следующий сегмент — какому из соседних маршрутизаторов переслать пакет. Это решение принимается каждым маршрутизатором для каждого пакета, поскольку Интернет — это сеть коммутации пакетов, а не каналов.

Картину глобальной связности маршрутизатор создает самостоятельно, обмениваясь информацией о топологии сети со своими соседями — какие сети подключены непосредственно к маршрутизатору, а какие достижимы через другие соседние узлы. Для обмена этой информацией применяются протоколы маршрутизации, например BGP, который используется в глобальной инфраструктуре.

---

[1] RFC 827: Exterior Gateway Protocol (EGP), URL: https://www.rfc-editor.org/rfc/rfc827

Такая архитектура позволяет автоматически реагировать на изменения топологии Интернета — выход из строя каналов, отдельных узлов или целых сетей. Это также означает, что поток данных может вдруг изменить свой путь вследствие изменения топологии сети (например, выхода из строя одного из транзитных узлов). Все это делает глобальную инфраструктуру Интернета устойчивой и адаптивной.

За последние три декады своего существования глобальная система маршрутизации выросла неимоверно — от 15000 маршрутов NSFNET до более миллиона маршрутов (включая IPv6) в 2023 году (рис. 35).

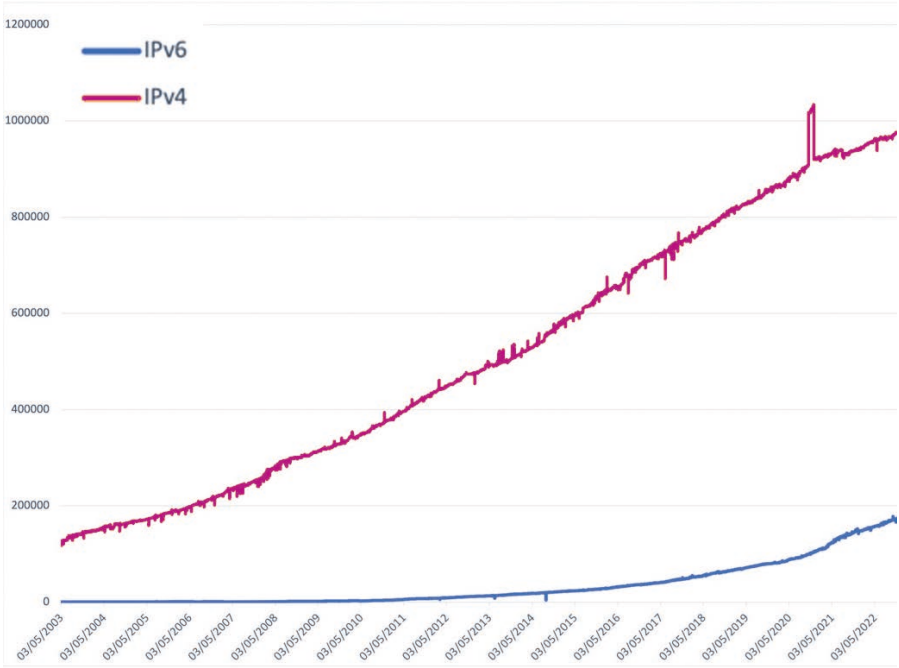

Рис. 35. Рост глобальной таблицы маршрутизации.
Источник данных: «BGP Analysis Reports» http://bgp.potaroo.net/index-bgp.html

Тем не менее, как это ни странно, основные протоколы и архитектура маршрутизации в Интернете почти не изменились. И удивительно, что, несмотря на столь стремительный рост, устойчивость и производительность системы по-прежнему соответствуют потребностям. Как заметил научный руководитель APNIC Джеф Хьюстон (Geoff Huston), это частично может быть объяснено тем, что «рост Интернета в большей степени означает увеличение плотности топологии, нежели ее размера». В качестве свидетельства он приводит относительно постоянную среднюю длину пути — среднее

число сетей, через которые проходит трафик от отправителя к получателю. На протяжении последнего десятка лет он составляет около четырех сегментов[2].

Это, впрочем, не означает, что в глобальной системе маршрутизации и ее основном протоколе BGP отсутствуют проблемы. Напротив, острота вопросов безопасности, обеспечения качества и эффективной конфигурации сети только усиливается.

Но об этом чуть позже. Сначала поговорим об истоках сегодняшней инфраструктуры и основных принципах маршрутизации в Интернете.

Междоменная маршрутизация: от EGP до BGP

Маршрутизация между сетями в Интернете осуществляется с помощью протокола BGP[3]. Суть его в том, что каждая сеть получает от своих соседей — пиров — информацию о связности, а именно — через какую цепочку сетей доступен конкретный префикс. Каждая сеть обрабатывает эту информацию в соответствии с собственной политикой, выбирая для каждого доступного префикса наилучший путь. Выбор этот основан на нескольких параметрах, основным из которых является длина пути. Таким образом, сеть формирует свое «видение» Интернета — и, в свою очередь, также сообщает о нем своим пирам.

Здесь следует отметить, что глобальная маршрутизация осуществляется не между отдельными компьютерами или маршрутизаторами, а между сетями, точнее, доменами маршрутизации. Эти домены (отсюда термин «междоменная маршрутизация» — Inter-Domain Routing, IDR) также называются автономными системами (Autonomous System, AS). Они представляют собой совокупность узлов, находящихся под единым административным контролем и имеющих согласованную политику маршрутизации. Этот аспект является одним из фундаментальных архитектурных принципов Интернета. Он позволяет эффективно разделить глобальную систему маршрутизации на области с четким внутренним контролем, относительно высокой степенью безопасности — и на «межобластное» пространство, основанное на кооперации и взаимодоверии.

Протокол BGP пришел на смену раннему протоколу маршрутизации EGP (Exterior Gateway Protocol), предложенному в 1982 году, когда тогдашний Internet (Catenet) начал расширяться, задействовав не только ARPANET, но и другие опорные сети. Добавление все новых и новых сетей несло в себе риск несовместимости и нестабильности, вызванных различными версиями протокола маршрутизации. Также в новой топологии появились транзитные сети — они передавали трафик, источник и получатель которого находились в других сетях. Протокол EGP

---

[2] См. статью «Update on AS Path Lengths Over Time» https://labs.ripe.net/Members/mirjam/update-on-as-path-lengths-over-time

[3] Border Gateway Protocol, https://ru.wikipedia.org/wiki/Border_Gateway_Protocol

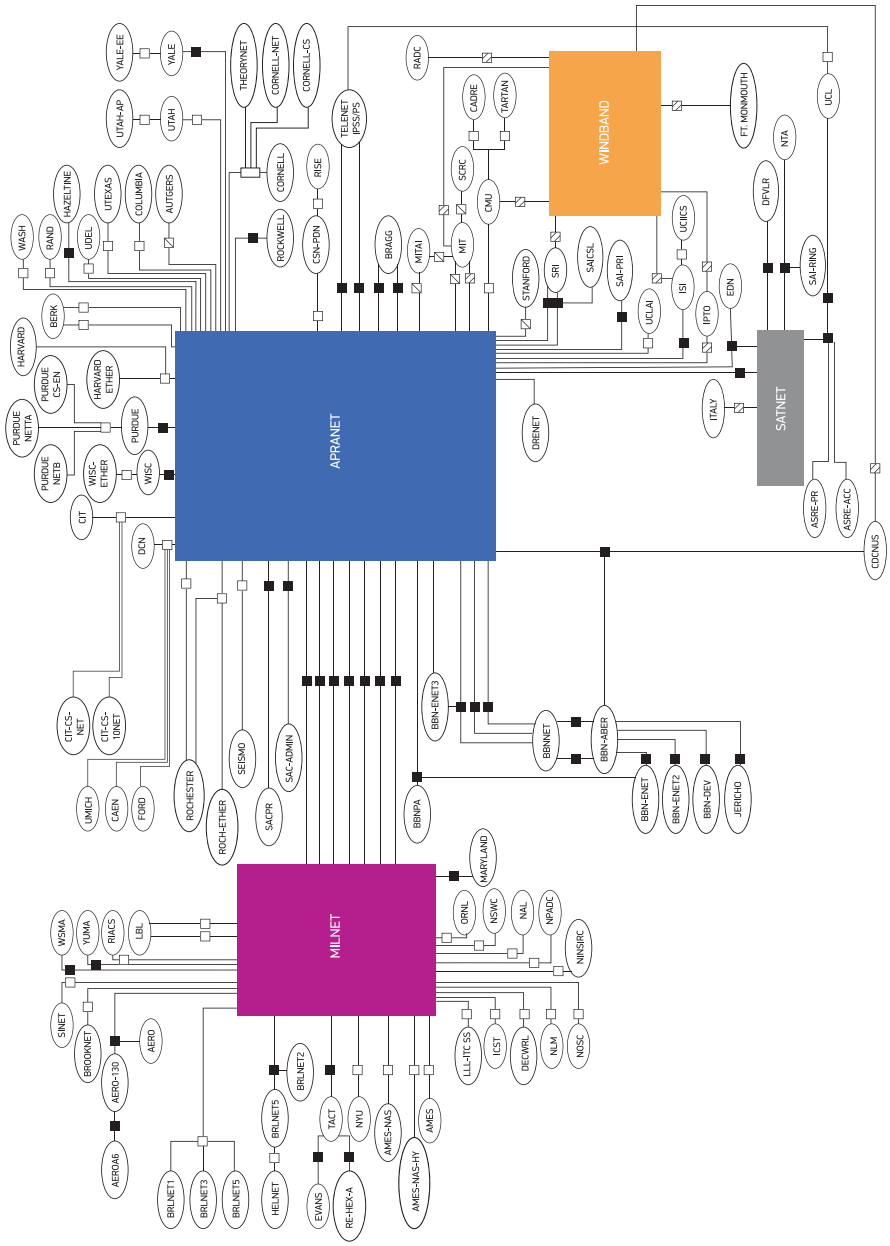

Рис. 36. Карта Интернета образца начала 1986 г.: небольшие квадраты — маршрутизаторы, овалы — сети, а крупные прямоугольники — опорные сети.
Источник: материалы второго совещания IETF в апреле 1986 г. (http://www.ietf.org/proceedings/02.pdf)

предусматривал строго иерархическую топологию и отсутствие колец, что хорошо соответствовало топологии Сети того времени. Топология тогдашней Сети показана на рис. 36.

В 1984 году NSF начал программу по созданию научно-исследовательских суперкомпьютерных центров, к которым могли бы иметь доступ американские ученые. Важной частью программы было создание высокоскоростной сети (NSFNET), связывающей эти суперкомпьютерные центры, а также все существовавшие региональные и академические сети. Сеть NSFNET, соединившая в 1986 году шесть суперкомпьютерных центров в США, являлась единым административным доменом с единой опорной сетью. Эта сеть представляла собой логическую точку обмена трафиком и, по существу, была в начале 1990-х опорной сетью Интернета. NSFNET также обменивалась трафиком с другими опорными сетями, такими как научная сеть NASA или ARPANET. В качестве протокола маршрутизации на начальном этапе было решено использовать протокол EGP.

Изначально предполагалось, что сеть будет иметь древовидную иерархическую топологию, напоминающую ARPANET, но на практике сети часто подключались к нескольким региональным сетям. Поскольку EGP не предоставлял необходимых метрик, на уровне опорной сети выбрать оптимальный маршрут было проблематично, и это иногда приводило к потере связности. Для решения проблемы было предложено использование так называемых политик маршрутизации — набора правил, которые сеть определяет для обмена маршрутизационной информацией с другими сетями. В данном случае каждой сети предлагалось[4] описать, с какими региональными сетями она связана, какой линк является основным, а какие — вторичными. Таким образом опорная сеть сможет использовать эту дополнительную информацию для принятия решений. Однако недостатком такой методики была плохая масштабируемость. Когда в конце 1980-х началось значительное расширение сети, назрела необходимость внедрения нового протокола.

BGP был разработан как для работы в иерархических сетях, таких как изначальная сеть NSFNET, так и в сетях с неиерархической топологией, когда сети одного уровня иерархии — пиры — могут быть непосредственно соединены друг с другом. Благодаря этому фундаментальному изменению BGP в дальнейшем обеспечил развитие Интернета как коллекции независимых, различным образом соединенных друг с другом сетей.

Первая версия протокола была стандартизована в 1989 году, хотя к тому времени он уже использовался в некоторых сетях. Согласно модели BGP каждый узел обменивается со своими соседями информацией о доступных путях к другим сетям, а именно о последовательности узлов, через которые должен быть передан трафик, чтобы достигнуть сети получателя. Таким образом,

[4] RFC 1092: EGP and policy based routing in the new NSFNET backbone, URL: https://www.rfc-editor.org/rfc/rfc1092

каждый узел имеет собственное представление о различных путях, но не о топологии Интернета в целом. Также, поскольку сети уже не могут рассматриваться как иерархические, недостаточно и информации о суммарной «стоимости» пути (как было в ранних протоколах маршрутизации), так как это может приводить к возникновению циклов. Чтобы избежать данной проблемы, каждый анонсируемый маршрут имеет специальный атрибут — AS_PATH, который содержит последовательность всех сетей, через которые этот анонс был передан. Если сеть обнаружит себя в списке AS_PATH полученного маршрута, это свидетельствует о цикле и такой маршрут должен быть отброшен.

Итак, отдельные сети, находящиеся под единым административным контролем и имеющие согласованную политику маршрутизации, обеспечивают отсутствие внутренних циклов. А значит, межсетевую и внутрисетевую маршрутизацию можно рассматривать независимо. Соответственно, в контексте межсетевой маршрутизации такие сети могут рассматриваться как метаузлы, или автономные системы (AS).

С одной стороны, данный подход позволил значительно упростить глобальную систему маршрутизации и обеспечить масштабируемость Интернета, с другой — он был основан на транзитивном доверии между взаимодействующими сетями.

Доверие — интересная вещь. В случае протокола BGP оно радикально упрощает взаимодействие между сетевыми операторами и, как следствие, систему маршрутизации в целом. Это, безусловно, явилось одним из факторов, обеспечивших бурное развитие Интернета. С другой стороны, доверие открывает существенные возможности для игроков не по правилам. Атаки «захват префикса» и перехват трафика случаются в Интернете постоянно, хотя прозрачность системы и сотрудничество между операторами играют существенную положительную роль. Об этом мы поговорим в следующих разделах.

BGP: принятие решений

В модели BGP каждая сеть, или автономная система, обменивается информацией об известных ей маршрутах с соседними сетями (в английской терминологии используются термины BGP neighbours, adjacent networks — смежные сети). Эта информация в терминах BGP называется NLRI (Network Layer Reachability Information, дословно — информация досягаемости сетевого уровня) и содержит префикс и его длину, что необходимо для поддержки CIDR. Например, NLRI/24, 198.51.100 указывает на маршрут к устройствам с IP-адресами в диапазоне 198.51.100.0–198.51.100.255.

Помимо NLRI в сообщении BGP (BGP Update) также содержатся так называемые атрибуты — параметры, связанные с анонсированным NLRI, которые используются при выборе «лучшего» маршрута и реализации политики маршру-

тизации. Дело в том, что маршрут к определенной сети, или NLRI, может быть получен от разных сетей, а в задачу BGP входит выбор «лучшего» из них. Слово «лучший» не случайно взято в кавычки — критерии и предпочтения в отношении маршрутов и потоков трафика определяются экономическими соображениями, а также топологией, распределением трафика и т.п. Они формируются в каждой автономной системе независимо. Обязательными и «общеизвестными» являются три атрибута: ORIGIN, AS_PATH и NEXT_HOP, — они присутствуют в любом анонсе BGP, их должны понимать все маршрутизаторы. Все три атрибута содержат информацию о пути маршрута:

1. **ORIGIN** — данные о том, как маршрут попал в глобальную систему BGP (ниже мы обсудим роль этого атрибута при выборе маршрута).
2. **AS_PATH** — список номеров автономных систем, через которые был передан этот маршрут.
3. **NEXT_HOP** — IP-адрес следующего маршрутизатора, которому нужно направить трафик, адресованный анонсированному NLRI. Обычно это маршрутизатор, от которого был получен анонс.

Говоря о процессе обработки маршрутов, стоит отметить, что анонсирование маршрута соседней сетью предполагает обязательство доставить данные, адресованные получателям этой сети. Сеть не контролирует, какие анонсы она получает от соседей, но она может проводить их фильтрацию и модификацию для влияния на стандартный процесс принятия решений BGP. Какие обязательства сеть готова взять на себя и какие сети-соседи использовать для передачи данных? Это и определяет политика маршрутизации, на которой мы остановимся в следующем разделе.

Процесс можно разделить на несколько этапов (см. рис. 35).

- Этап обработки входящих анонсов, полученных от соседних автономных систем. Для каждой автономной системы, с которой установлен сеанс BGP, маршрутизатор поддерживает отдельную базу маршрутов RIB-In (Routing Information Base). Если анонс вызывает изменение в этой базе, тогда обработка продолжается. Перед передачей полученных новых маршрутов (или информации об удалении существующих) процессу выбора маршрутов BGP для каждого соседа применяются правила, установленные политикой импорта. О политике маршрутизации мы поговорим в следующем разделе, здесь же скажем, что правилом может быть, например, фильтрация определенных маршрутов или изменение определенных атрибутов для влияния на процесс выбора.
- Этап выбора лучшего маршрута и его установка в локальную таблицу маршрутизации и таблицу передачи — Forwarding Information Base (FIB), которая отвечает за коммутацию пакетов с одного интерфейса маршрутизатора на другой на низком уровне. При выборе лучшего маршрута принимаются во внимание новые маршруты, полученные от соседей и прошедшие обработку, определенную политикой импорта.

- Этап обработки исходящих анонсов, передаваемых соседним автономным системам, об изменениях в локальной таблице маршрутизации. Эти изменения также сначала обрабатываются в соответствии с политикой экспорта, которая может, например, указывать, что некоторым соседям анонсы не должны отправляться вообще.

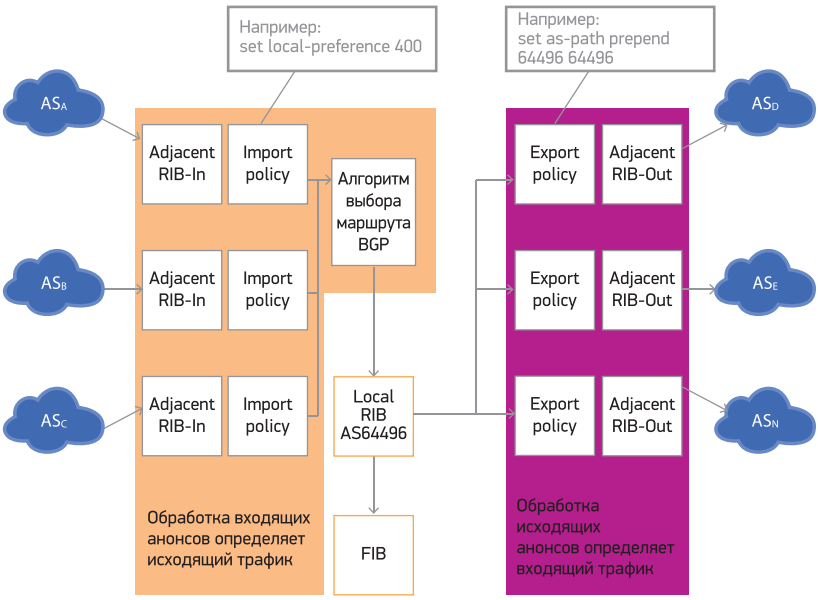

Рис. 37. Схема обработки анонсов BGP-маршрутизатором.

Алгоритм выбора лучшего маршрута к определенной сети (разумеется, в случае, когда маршрутизатором получено более одного маршрута) описан в RFC 4271 «A Border Gateway Protocol 4 (BGP-4)»[5]. В таблице ниже приведены шаги выбора в порядке приоритета. Если результатом какого-либо шага является единственный маршрут, процесс останавливается. Верно и обратное — пока у BGP есть выбор, процесс будет продолжаться, сравнивая более низкоприоритетные параметры маршрутов.

Таблица 6. Шаги выбора маршрута

Приоритет	Метрика	Правило
1.	LOCAL_ PREF	Выбирается маршрут (NLRI) с более высоким значением атрибута LOCAL_PREF. Этот атрибут устанавливается в рамках политики импорта и используется только внутри автономной системы для информирования других граничных маршрутизаторов о приоритете того или иного маршрута.
2.	AS_PATH	Выбирается маршрут с наименьшей AS_PATH. Эта метрика содержит последовательность всех сетей, через которые он был передан. Соответственно, BGP выбирает кратчайший маршрут. В общем случае AS_PATH является основной метрикой принятия решений.

[5] RFC 4271: A Border Gateway Protocol 4 (BGP-4), URL: https://www.rfc-editor.org/rfc/rfc4271#section-9.1.2

Приоритет	Метрика	Правило
3.	ORIGIN	Выбирается маршрут с наименьшим значением этого атрибута. ORIGIN указывает, как маршрут изначально появился в системе BGP, и является своего рода рудиментом со времени, когда Интернет перешел от протокола EGP к BGP. Возможных значений этого атрибута три: ● (0) IGP или internal, когда сеть непосредственно подключена к маршрутизатору, анонсирующему ее в BGP; ● (1) EGP — если маршрут был получен от EGP (это значение может быть установлено искусственно, поскольку EGP давно не используется в Интернете); ● (2) Incomplete — указывает, что маршрут был получен от других протоколов маршрутизации, поэтому реальный путь (за пределами AS_PATH) неизвестен.Выбор происходит в порядке 0–1–2. Обычно этот атрибут не привлекает особого внимания операторов, но при желании может использоваться для влияния на выбор маршрута другими автономными системами, поскольку он обязательно присутствует во всех анонсах BGP.
4.	MED	Выбирается маршрут с наименьшим значением этого атрибута. MED (Multi-exit discriminator, дискриминатор нескольких выходов) используется для определения политики передачи трафика между сетями, связанными в нескольких географически распределенных точках.
5.	Откуда получен маршрут	Маршрут, полученный от соседей (eBGP), предпочитается маршруту, полученному от других граничных маршрутизаторов (iBGP).
6.	Минимальный внутренний маршрут	Выбирается маршрут с минимальным путем по внутренней инфраструктуре до узла NEXT_HOP — маршрутизатора, передавшего анонс.
7.	Номер AS	Если маршруты были получены от внешних сетей, выбирается маршрут от автономной системы с наименьшим номером.
8.	ID внутреннего маршрутизатора	Если маршруты были получены от граничных маршрутизаторов своей же сети, выбирается маршрут, переданный маршрутизатором с «наименьшим» IP-адресом.

Последние два правила не имеют никакого практического смысла и используются для предсказуемого выбора маршрута, если все остальные правила привелитаки к ничьей.

Связность: пиры, клиенты и политика маршрутизации

Хотя BGP и определяет стандартный алгоритм выбора маршрута, который мы только что рассмотрели, фактически выбор делается на основе политики маршрутизации, принятой данной автономной системой. Другими словами, администратор сети может явно определить правила выбора маршрута в определенных условиях. Например, изначально политика маршрутизации в NSFNET разделяла коммерческий и некоммерческий трафик. Важно представлять, что этот выбор основан на информации, полученной сетью от ее соседей, или пиров. При этом не существует общей «эталонной» топологии Интернета — каждая автономная система вырабатывает свою собственную картину мира, которую, в свою очередь, транслирует своим пирам.

Другими словами, политика маршрутизации позволяет сети менять стандартный процесс выбора маршрута BGP. Во многих случаях эти изменения касаются собственного выбора (политика импорта), но также используются приемы для влияния на процесс выбора лучшего пути других сетей (политика экспорта). Для определения политики существенное значение имеет тип отношений между взаимодействующими сетями.

Здесь можно выделить два основных типа отношений между сетями: клиент-провайдер (транзит) и пиринг (peering).

Отношения клиент-провайдер предусматривают, что провайдер предоставляет услуги транзита сети клиента. Это означает, что клиент получает доступ не только к сетям провайдера, но и к другим, внешним сетям. Как правило, под внешними сетями подразумевается глобальный Интернет. Обычно оплата клиента за оказание таких услуг зависит от пропускной способности канала и его фактической загрузки.

В случае пиринга две сети (пиры) договариваются об обмене трафиком между собственными сетями, включая своих клиентов. Обычно пиринг не предусматривает взаиморасчетов, поскольку такие отношения заключаются между равноценными сетями, когда оба партнера получают примерно одинаковую выгоду от обмена трафиком. Основными преимуществами пиринга являются, конечно, облегчение нагрузки (и, соответственно, расходов) на провайдера транзита, а также повышение устойчивости за счет более богатой связности.

Это, конечно же, упрощенная картина. Существуют и другие разновидности политик обмена трафиком — платный пиринг, пиринг провайдера контента и т.п. Все они определяются в первую очередь экономическими соображениями.

Глобальная система маршрутизации обладает удивительной способностью к самоорганизации — каждая из сетей формирует свою политику независимо, и тем не менее, в результате обеспечивается глобальная связность. Это еще более удивительно, учитывая, что сегодняшний Интернет имеет не такую иерархическую структуру, как, скажем, NSFNET (см. рис. 36). Правда, хотя в Интернете и нет единой опорной сети, существует дюжина крупнейших сетей, таких как AT&T, Level3, Deutsche Telecom, которые играют ключевую роль в обеспечении глобальной связности. Эти сети еще называют сетями первого уровня, или сетями Tier-1. Все остальные сети Интернета являются непосредственными или опосредованными клиентами сетей Tier-1. У сети Tier-1 нет провайдера транзита — обмен трафиком между собой они осуществляют на основе пиринговых отношений. На этом уровне и формируется глобальная связность. Транзит и пиринг, помимо взаимного соглашения — контракта в случае транзита и чаще всего устной договоренности в случае пиринга, — обеспечиваются техническими средствами. С каждым из типов связана определенная политика маршрутизации и, соответственно, технические средства ее реализации.

Говоря о технических средствах, необходимо упомянуть о специальном языке описания политики маршрутизации — RPSL (Routing Policy Specification Language). Синтаксис этого языка определен в документе RFC 2622[6] с расширениями для поддержки адресации IPv6 и мультикаст, описанными в RFC 4012[7]. Возможность описания и публикации политики нужна, чтобы определять и исправлять проблемы маршрутизации в глобальном масштабе. RPSL также позволяет генерировать конфигурационные файлы для маршрутизаторов, осуществляющих эту политику.

Не будем вдаваться в глубины этого языка, приведем лишь несколько примеров политик в стиле RPSL. Напомним, что политики импорта влияют на потоки исходящего трафика, а политики экспорта — входящего.

Транзит: AS1-провайдер, AS2клиент, AS2-CUSTOMERS — все клиенты AS2.

```
aut-num:    AS2
import:     from AS1 accept ANY
export:     to AS1 announce AS2 AS2-CUSTOMERS

aut-num:    AS1
import:     from AS2 accept AS2 AS2-CUSTOMERS
export:     to AS2 announce ANY
```

Пиринг: AS1 и AS2 являются пирами.

```
aut-num:    AS2
import:     from AS1 accept AS1 AS1-CUSTOMERS
export:     to AS1 announce AS2 AS2-CUSTOMERS

aut-num:    AS1
import:     from AS2 accept AS2 AS2-CUSTOMERS
export:     to AS2 announce AS1 AS1-CUSTOMERS
```

Многие сети одновременно играют все три роли: провайдера — для более мелких сетей, клиента — для получения транзита и глобальной связности и пира — для обмена трафиком с равноценными сетями. Обычно маршруты, полученные от клиентов, предпочитаются маршрутам от пиров, а маршруты от провайдера транзита обычно имеют наименьший приоритет. Это разумно: хотя ваш клиент может также анонсироваться и пиром, все-таки предпочтительнее

---

6 RFC 2622: Routing Policy Specification Language (RPSL), URL: https://www.rfc-editor.org/rfc/rfc2622

7 RFC 4012: Routing Policy Specification Language next generation (RPSLng), URL: https://www.rfc-editor.org/rfc/rfc4012

посылать трафик клиенту напрямую. А для доступа к сетям пира неразумно пользоваться платным транзитом, поэтому транзитные анонсы и стоят последними при выборе наилучшего маршрута.

Однако эти правила не являются стандартными в выборе маршрута BGP: определения «клиент», «пир» или «провайдер» обозначают деловые отношения и не имеют смысла в контексте протокола[8]. Такие правила должны быть реализованы политикой импорта. Помните, мы говорили об атрибуте LOCAL_PREF, позволяющем изменять предпочтение маршрута? Этот атрибут не передается между автономными системами, поэтому он полезен только для принятия собственных решений и оповещения о них других граничных маршрутизаторов сети.

Если AS1 получает транзит от AS4, осуществляет пиринг с AS3 и является провайдером для AS2, то политика импорта этой сети может выглядеть следующим образом:

```
aut-num:    AS1
import:     from AS2 action pref=300; accept AS2 AS2-CUSTOMERS
import:     from AS3 action pref=200; accept AS3 AS3-CUSTOMERS
import:     from AS4 action pref=100; accept ANY
```

Отсутствие контроля за выполнением этих политик на уровне конфигурации фильтров import и export маршрутизатора (см. рис. 36) может привести к аномалиям. Например, вследствие ошибки конфигурации пир может стать провайдером транзита. Это может привести к неоптимальным потокам трафика и потере качества. Гораздо более серьезные последствия могут возникнуть, если вашим провайдером вдруг станет клиент. Получение от него маршрута по умолчанию («default», о/о) приведет именно к такой ситуации, поскольку маршруты, полученные от клиента, всегда имеют приоритет. Наиболее вероятно, что последствием будет отказ в обслуживании вследствие перегрузки инфраструктуры сети клиента. Но «утечка» может быть и менее заметной, например, если речь идет всего о нескольких маршрутах, принадлежащих другим сетям. Такие ситуации получили название «утечка маршрута» (route leak), и они могут иметь серьезные последствия в плане безопасности, поскольку позволяют клиенту «вставить» свою сеть между вашими пользователями и внешними сетями, открывая широкие возможности для атак типа MITM и прослушивания трафика. Представьте, что утечке подвергнется сеть сайта онлайн-платежей

[8] Недавно принятый стандарт RFC9234 "Route Leak Prevention and Detection Using Roles in UPDATE and OPEN Messages" (https://www.rfc-editor.org/rfc/rfc9234) позволяет зафиксировать характер взаимоотношений между сетями на этапе создания BGP-сессии. Впоследствии это может быть использовано для контроля правильности политики маршрутизации. Более подробно мы поговорим об этом в разделе «Безопасность системы маршрутизации».

или сайта социальной сети. Мы вернемся к этой теме в разделе «Безопасность системы маршрутизации».

Соблюдение правил, диктуемых характером взаимоотношений между сетями, обеспечивает маршрут с характеристикой, получившей название valley-free («не пересекая долины»). Маршрут является valley-free, если он состоит из трех последовательных участков, каждый из которых может быть нулевым: движение в направлении клиент-провайдер, максимум один пиринговый линк и, наконец, движение в направлении провайдер-клиент. В этом случае клиенты остаются клиентами, пиры — пирами, а провайдеры предоставляют заранее оговоренный транзит.

Чтобы получить иллюстрацию этого факта, взгляните на рис. 38. Маршруты AS_2-AS_1-AS_C-AS_{10} и AS_5-AS_3-AS_7-AS_6 являются «правильными» с точки зрения соблюдения политик. В то же время маршруты AS_1-AS_5-AS_3, AS_7-AS_8-AS_9 и AS_B-AS_8-AS_9 явно не соответствуют предполагаемым отношениям между сетями. Вы заметите, что все эти маршруты содержат «спуск в долину» или движение в «долине» через два и более пиринговых линка.

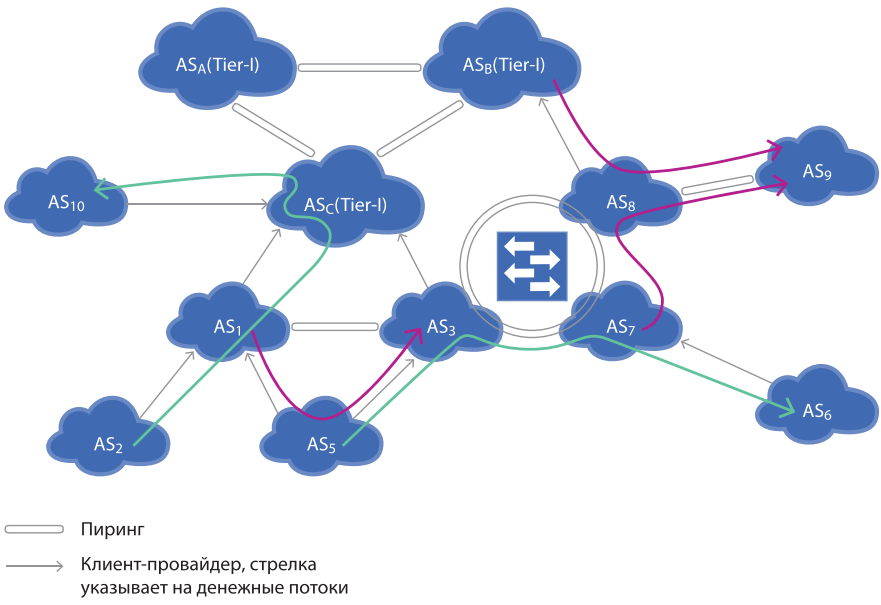

Рис. 38. Структура сегодняшнего Интернета. «Правильные» и «неправильные» маршруты между связанными сетями. Красными линиями обозначены маршруты, нарушающие принцип valley-free (непересечения долины).

Безопасность системы маршрутизации

Протокол маршрутизации BGP по существу является нервной системой Интернета. Несмотря на свою фундаментальную значимость, протокол BGP основан на доверии между соединенными сетями, ведь полученная от них информация принимается за чистую монету. Более того, это доверие обладает транзитивным свойством: пиры доверяют своим соседям, те, в свою очередь, — своим, и в итоге все доверяют всем.

Другими словами, BGP позволяет «лгать». И эта ложь, если не поставить дополнительных заслонов, будет распространяться по всему Интернету. Она может быть следствием ошибок конфигурации или умышленным подлогом. Последствия ее могут быть невинны и незаметны, а могут перерасти в атаку, затрагивающую все системы и сервисы.

Разумеется, сеть вольна не доверять полученной информации и осуществить дополнительную проверку. Но, к сожалению, собственно протокол BGP здесь вряд ли поможет, поэтому необходимо прибегать к дополнительным средствам, о которых мы и поговорим в этом разделе.

Как можно атаковать систему маршрутизации

В информационном документе IETF RFC 4593[9] обсуждаются потенциальные угрозы системы маршрутизации, а RFC 4272[10] подробно указывает на уязвимые места протокола BGP. Вот эти уязвимости:

- отсутствие внутреннего механизма, обеспечивающего сильную защиту целостности, свежести и аутентичности сообщений BGP, которыми обмениваются сети-пиры друг с другом;
- отсутствие механизма для проверки прав автономной системы или сети анонсировать префикс;
- отсутствие механизма для проверки подлинности атрибутов пути, анонсированных сетью-пиром.

Векторы атаки на систему маршрутизации представлены на рис. 39. Первая проблема имеет отношение к защите канала между пирами и часто решается локальными средствами. Рабочая группа IETF KARP[11] предлагает продвинутые решения в этой области. Две остальные группы уязвимости имеют более существенное значение в глобальном масштабе.

---

9 RFC 4593: Generic Threats to Routing Protocols, URL: https://www.rfc-editor.org/rfc/rfc4593

10 RFC 4272: BGP Security Vulnerabilities Analysis, URL: https://www.rfc-editor.org/rfc/rfc4272

11 https://datatracker.ietf.org/wg/karp

Можно выделить несколько общих типов атаки. Несмотря на различие целей и конечного эффекта, механизм атаки принципиально строится на том, что в атакуемой сети создается искаженная картина топологии Интернета. Затем эта искаженная картина транзитивно распространяется по всей Сети.

Создание «черных дыр». Цель этой атаки — отрезать сеть или несколько сетей от всего Интернета или его части. Весь трафик, имеющий отношение к этим сетям, перенаправляется и затем отбрасывается. В результате все сервисы, предлагаемые данными сетями, становятся недоступными для пользователей. Основным результатом атак этого типа является отказ в обслуживании (Denial of Service, DoS).

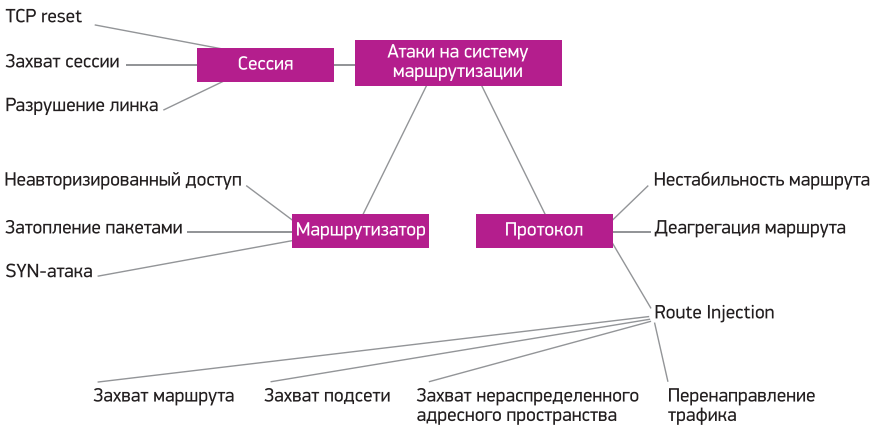

Рис. 39. Атаки на систему маршрутизации.

Перенаправление. В этом случае трафик, предназначенный для одной сети, перенаправляется в другую. Часто эта сеть находится в руках атакующего и маскируется под атакуемую сеть — например, с целью получения секретной информации. Также перенаправление может быть использовано для проведения злоумышленниками определенных краткосрочных акций, например, рассылки спама. После этого такая сеть, или ее фантом, исчезает. Часто злоумышленниками используется нераспределенное или давно не используемое адресное пространство.

Перехват. Эта атака похожа на предыдущую, только после прохождения по сети-перехватчику трафик возвращается в нормальное русло и попадает к получателю. Из-за этого такую атаку труднее обнаружить. Целью обычно является «подслушивание» или модификация передаваемых данных.

Нестабильность. Нестабильность в глобальной системе маршрутизации может быть вызвана частыми изменениями в анонсировании конкретной сети (попеременное анонсирование и аннулирование). Цель — «демпфирование» маршрутов данной сети провайдерами и, как следствие, блокирование связности.

Фабрикация адреса источника трафика. В этом случае непосредственно система маршрутизации не подвергается атаке, но данный метод широко используется в так называемых рефлекторных атаках, о которых мы говорили в главе 2. Обратный трафик (например, ответы на изначальные запросы) направляется не к истинному источнику, а к получателю, чей адрес был сфабрикован. Как правило, такие атаки используют протокол без установления соединения UDP[12] и основаны на эффекте усиления, когда небольшие запросы от многих источников порождают ответы значительно большего размера. Одной из критических систем, в основном использующей UDP и подверженной атакам такого рода, является DNS.

Раз нет механизмов проверки подлинности полученной информации, значит атакующий может повлиять на маршрутизацию трафика, относящегося к той или иной сети, в глобальном масштабе. Наиболее распространенными является захват префикса, или маршрута (prefix hijacking, route hijacking), когда префикс какой-либо сети анонсируется атакующим — и трафик, предназначенный этой сети, перенаправляется в сторону атакующего (см. рис. 40).

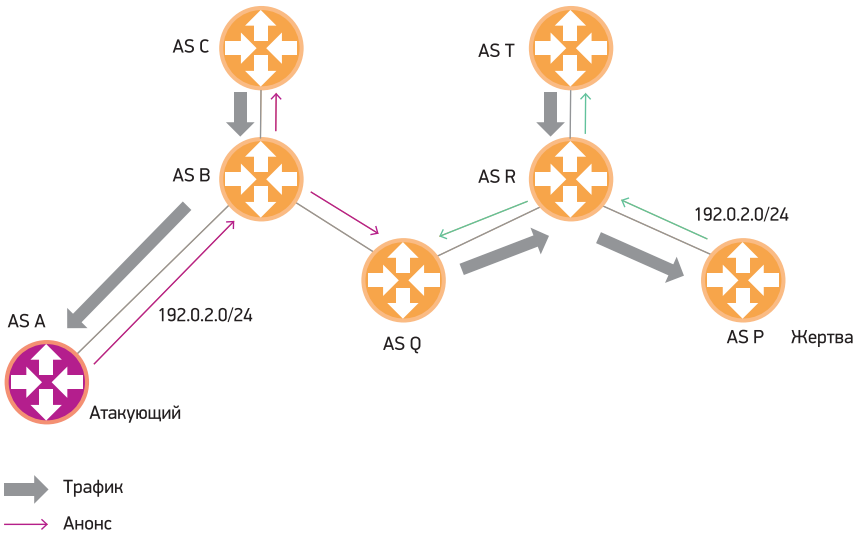

Рис. 40. Захват префикса. Атакующая сеть AS A анонсирует префикс, принадлежащий сети AS P. В результате трафик в части Интернета перенаправляется к атакующей сети.

Эта атака имеет несколько вариантов:

- захват маршрута, когда сеть анонсирует не принадлежащее ей адресное пространство в качестве сети-источника. При выборе маршрута BGP предпочтет более короткий путь, измеряемый числом сетей между источником

[12] User Datagram Protocol, http://ru.wikipedia.org/wiki/Udp

и получателем маршрута. Таким образом, захваченный маршрут будет конкурировать с истинным;

- захват подсетей, когда анонсируются более специфичные префиксы. При выборе маршрута BGP предпочитает тот, который указывается более специфичным префиксом, и таким образом атакующий выигрывает, несмотря на топологическую удаленность. В случае отсутствия конкурирующих префиксов такого же размера захват оказывает глобальный эффект;
- захват нераспределенного или неиспользуемого адресного пространства. В этом случае анонсируемый префикс не встречает конкуренции и имеет высокие шансы распространения по всему Интернету. В то же время очевидные недопустимые префиксы, так называемые bogons, как правило, фильтруются провайдерами.

Классическим примером атаки захвата префикса является захват сайта YouTube в феврале 2008 года, когда оператор Pakistan Telecom (AS17557) начал несанкционированное анонсирование части адресного пространства, используемого YouTube (AS36561), а именно более конкретного (more specific) префикса 208.65.153.0/24. Один из транзитных провайдеров Pakistan Telecom, PCCW Global (AS3491), проанонсировал данный маршрут далее, в глобальный Интернет, что привело к перенаправлению трафика YouTube в глобальном масштабе.

Топология связности YouTube уже спустя две минуты выглядела следующим образом:

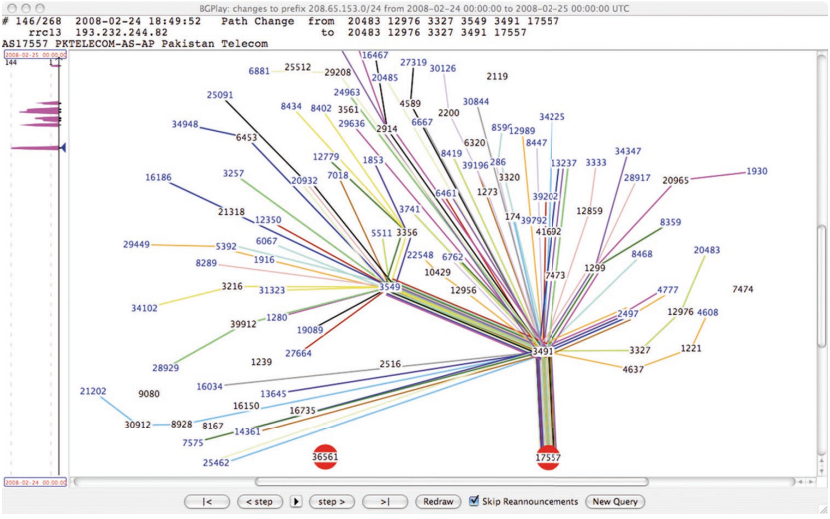

Рис. 41. Топология связности YouTube (AS36561) после начала анонсирования префикса Pakistan Telecom (AS17557).
Источник: «YouTube Hijacking: A RIPE NCC RIS case study» (http://www.ripe.net/internet-coordination/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study)

Как видно, весь трафик, предназначенный YouTube, был перенаправлен в сеть Pakistan Telecom. Этот трафик, скорее всего, представлял собой обрывки сеансов TCP, начатых с реальным сайтом YouTube, а также попытки начать новые сеансы; вероятно, он попросту отбрасывался сетью Pakistan Telecom. Для пользователей YouTube это выглядело как недоступность ресурса.

Причиной данного эксцесса явилось требование правительства Пакистана заблокировать доступ к враждебному сайту внутри страны. Однако в результате была создана типичная «черная дыра», что и привело к глобальному сбою в работе служб YouTube.

Последствия атак этого типа могут быть различными. Захват маршрута приводит к перетягиванию трафика, предназначенного «захваченной» сети, который, как правило, затем отбрасывается. То есть происходит DoS-атака на все сервисы сети. Такая атака может также быть использована для краткосрочной генерации трафика, например, для рассылки спама. В более изощренном виде захват маршрута может быть направлен на захват некоторого информационного ресурса, например веб-сайта, когда пользователям демонстрируется подложный сайт. В этом случае даже защита DNSSEC будет бессильна. А учитывая относительную простоту получения сертификата TLS, такая атака может иметь серьезные последствия для пользователей — например, кражу данных по кредитным картам.

Атака Пилосова (Pilosov)

Атака на YouTube имела значительные видимые последствия, она получила широкую огласку и резонанс в сетевом сообществе. Однако ряд атак могут проходить почти незаметно — но приводить к не менее серьезным, а то и более значительным последствиям.

Речь идет о перехвате трафика, незаметном как для отправителя и получателя трафика, так и для большинства других участников обмена информацией. Целью может быть, например, перлюстрация данных, которыми обмениваются определенные сети или пользователи. Злоумышленник может также попытаться модифицировать эти данные.

Возможность и простота организации такой атаки были описаны на конференции DEFCON в августе 2008 года Алексом Пилосовым (Alex Pilosov) и Антоном Капелла (Tony Kapella). Они продемонстрировали, что:

- практически любой префикс может быть захвачен без нарушения сквозной связности;
- это можно сделать очень незаметно, замаскировав присутствие атакующего на пути следования трафика (он невидим для утилит типа traceroute, позволяющих получить список узлов, через которые передается трафик к получателю).

Суть атаки сводится к перехвату трафика стандартными методами (например, путем анонсирования атакующим более длинного префикса атакуемой сети, что делает анонс данной подсети более привлекательным с точки зрения BGP, как это произошло в случае с Pakistan Telecom). Далее трафик возвращается в прежнее русло. Для этого атакующая сеть может, например, установить статический маршрут (рис. 41). В результате трафик передается сети, являвшейся частью изначального пути передачи трафика. Далее передача трафика происходит абсолютно законным путем.

Для маскировки движения трафика атакующая сеть производит манипуляцию с параметром TTL (Time To Live) пакетов перехваченного трафика. Согласно протоколу, при передаче пакета от одного маршрутизатора к другому каждый узел сети уменьшает этот параметр на единицу. Не изменяя TTL при прохождении по атакующей сети, злоумышленники могут «замаскировать» этот участок, исключив таким образом атакующего из видимого пути передачи трафика. Для утилит типа traceroute участки пути в сети атакующего просто не попадают в список.

Атака Пилосова проиллюстрирована на рис. 42 и 43. Первый рисунок показывает состояние системы маршрутизации перед началом атаки, когда трафик от пользователя сети AS_{70} доставляется через AS_{60} получателю сети AS_{200}. На втором изображена связность сетей в процессе атаки. Хотя трафик перехватывается атакующей сетью AS_{100}, этот путь не отражается программой traceroute.

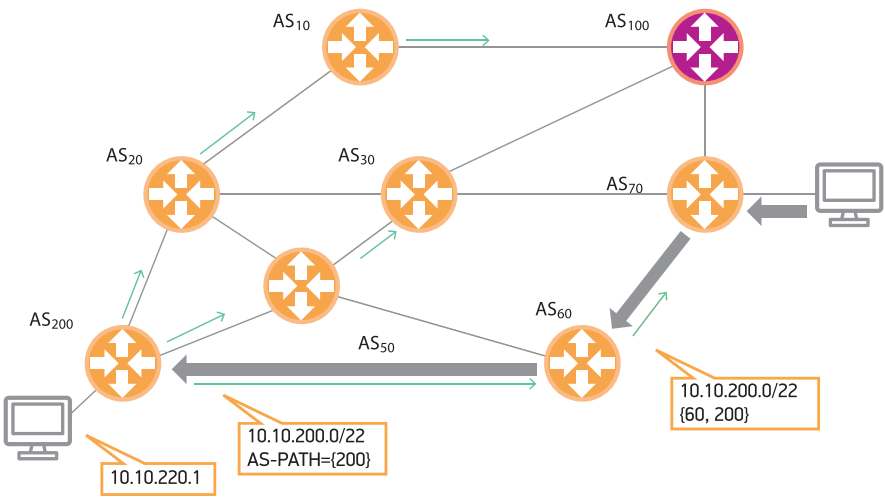

Рис. 42. Состояние системы маршрутизации перед началом атаки Пилосова.

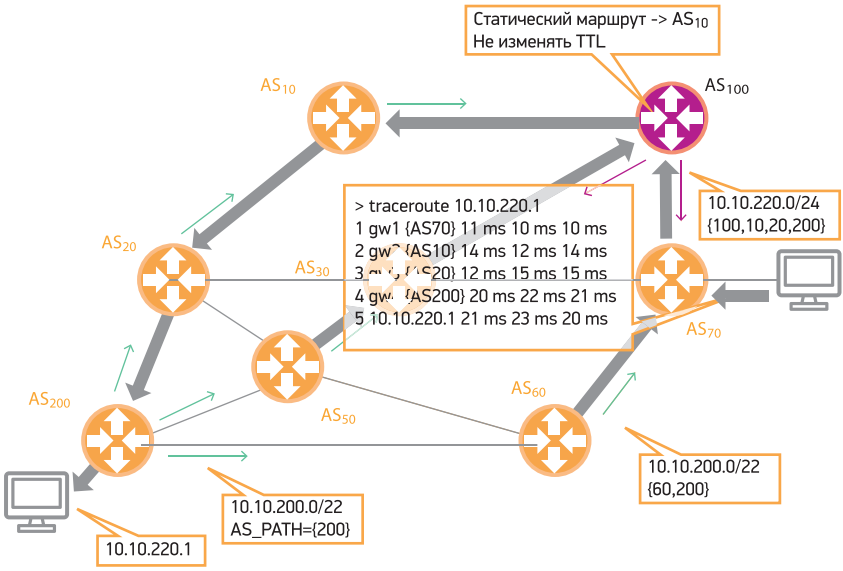

Рис. 43. Перехват трафика атакующей сетью (AS_{100}).

Упрощенной версией атаки Пилосова–Капеллы является простая «утечка маршрута» (route leak), о которой мы говорили в предыдущем разделе. Безусловно, в этом случае аномалия не маскируется, но в случае ограниченной утечки и умеренного трафика может оставаться незамеченной на протяжении долгого времени.

Очевидно, что глобальная система маршрутизации уязвима для всех перечисленных атак. Тем не менее, связанные с ними риски зачастую считаются незначительными и «принимаются» без дополнительных мер защиты. Отчасти это связано с тем, что многие из «атак» являются последствиями ошибок конфигурации, достаточно заметны за счет прозрачности системы и достаточно быстро разрушаются. Опытные операторы рассматривают эту проблему более серьезно; правда, в большинстве случаев учитываются только риски, связанные с непосредственным окружением — сетями, клиентами и пирами. Как правило, это отражается в практике фильтрации маршрутов собственных сетей-клиентов и установки максимального числа принимаемых маршрутов от пиров и сетей, предоставляющих транзит. Реже производится фильтрация маршрутов, полученных от пиров.

Существующая практика безопасности маршрутизации

Как вы, наверное, заметили, рассмотренные выше атаки являются нарушением предполагаемой политики маршрутизации. Одно дело определить политику, и совсем другое — обеспечить ее выполнение техническими средствами. Необходимо получить достоверные данные о правомерных маршрутах, полученных

от клиента, пира или провайдера. Например, если наша политика по отношению клиента AS_2 выглядит так:

```
aut-num:    AS1
import:     from AS2 accept AS2 AS2-CUSTOMERS
export:     to AS2 announce ANY
```

то как определить легитимные маршруты, которые могут быть анонсированы AS2 и ее клиентами? В самом протоколе BGP эта информация отсутствует, поэтому возникает необходимость во внешних источниках информации.

Итак, безопасность и надежность системы маршрутизации во многом зависят от возможности правильного ответа на вопросы:

- Является ли префикс, полученный в сообщении BGP, правомерным (то есть представляющим законно распределенное адресное пространство и право на его использование)?
- Является ли автономная система-отправитель сообщения BGP правомочным источником префикса?
- Соответствует ли атрибут AS_PATH, полученный в сообщении BGP, действительному пути, который прошло данное сообщение в сети Интернет?

К сожалению, дать правильные ответы на поставленные вопросы трудно ввиду отсутствия надежного источника информации. Но что же все-таки имеется в арсенале сервис-провайдера?

Интернет-регистратуры маршрутизации (IRR)

Частичную помощь в решении данной проблемы оказывают интернет-регистратуры маршрутизации (Internet Routing Registry, IRR). Суть их в следующем: сетевые операторы регистрируют в базе данных свою политику маршрутизации, а именно — с кем и как сеть взаимодействует. Также регистрируются и префиксы, которые сеть использует и анонсирует в Интернете. Для описания политик используется язык RPSL, о котором мы уже говорили. А инструментарий, наиболее известный из которых — IRRToolset[13], позволяет автоматизировать конфигурацию маршрутизации провайдера по данным IRR.

Но IRR отображают весьма неполную картину, так как регистрация данных в этих базах сугубо добровольная. Многие операторы не хотят морочить себе голову какими-то IRR, часть операторов не регистрируется там по причине нежелания разглашать свою политику. Те же, кто все же зарегистрировал свою политику, не всегда поддерживают актуальность данных. Проблема в том, что хотя эта деятельность служит на благо общего дела — безопасной системы маршрутизации, польза для самого провайдера не всегда ощутима.

[13] https://www.isc.org/othersoftware/#IRR

Неполнота и ненадежное качество данных, а также плохая масштабируемость — попробуйте-ка создать фильтры для всех префиксов, зарегистрированных в IRR! — существенно ограничивают применение интернет-регистратур для решения проблем безопасности глобальной маршрутизации.

В результате IRR имеют весьма ограниченное распространение и в основном используются для администрирования провайдером подключенных клиентов.

Whois

Можно воспользоваться более надежными базами данных распределения адресного пространства на уровне региональных интернет-регистратур (Regional Internet Registry, RIR). Эту информацию можно получить через соответствующий whois-сервер регистратуры, но иногда более практично использовать так называемые файлы статистики, доступные на ftp-сайте всех РИРов[14]. Например, интернет-ресурсы, распределенные RIPE NCC, представлены в отдельном файле[15].

Как вы можете заметить, число записей весьма внушительно. Также внушительным будет и список префиксов в конфигурации ваших граничных маршрутизаторов.

База данных IANA (Internet Assigned Number Authority, iana.org), например, для ресурсов IPv4[16], является более компактной, хотя и не содержит деталей — каждая запись имеет размер /8 в случае IPv4, а детализация для адресного пространства IPv6 и того меньше. Однако данный подход позволяет по крайней мере блокировать сети, использующие нераспределенные адресные ресурсы.

Сертификация номерных ресурсов и безопасность маршрутизации

Как мы уже говорили, существенные препятствия на пути обеспечения безопасной маршрутизации — недостаточная доступность и надежность данных, позволяющих оператору принять решение о достоверности анонсируемых маршрутов.

Система, построенная на основе RPKI, призвана решить эти проблемы. Ее фундамент — система открытых ключей (Public Key Infrastructure, PKI), элементами которой являются сертификаты интернет-ресурсов (CP). На основе этих сертификатов держатели ресурсов могут создавать криптографически заверенные объекты, например, ROA (Route Origin Authorization), указывающие на список автономных систем, которые могут являться источником определенного маршрута.

Во-первых, данные о распределенных номерных ресурсах предоставляются в стандартной форме цифровых сертификатов со стандартными расширениями (расширения X.509, о которых чуть позже, как раз содержат список ресурсов,

[14] Например, ftp://ftp.ripe.net/pub/stats

[15] ftp://ftp.ripe.net/pub/stats/ripencc/delegated-ripencc-latest

[16] http://iana.org/assignments/ipv4-address-space/ipv4-address-space.xml

привязанных к открытому ключу сертификата). Во-вторых, достоверность и свежесть данных может быть проверена с использованием криптографических средств третьими лицами. Как и в стандартном PKI, для проверки необходима конфигурация доверия только к одному сертификату — так называемые точки доверия (Trust Anchor, TA). В-третьих, сертификаты ресурсов могут использоваться их владельцами (держателями адресного пространства), например, для электронной авторизации определенных автономных систем для анонсирования данного адресного пространства. Таким образом, сертификаты выполняют функцию объектов route традиционных IRR.

Структура RPKI

Как и в случае традиционной PKI, в состав RPKI входят следующие компоненты (см. RFC 6480[17]):

Удостоверяющий центр сертификации (УЦС), задачи которого — выдача сертификатов и их отзыв. УЦС содержит две важные службы.

- Служба выдачи сертификатов (Certificate Authority, CA). Основной функцией CA является генерация и публикация сертификатов и списков аннулированных сертификатов. Эта функция, по существу, не меняется в RPKI, за исключением того, что CP содержат расширения, документирующие распределенные АИР.
- Служба регистрации (Registration Authority, RA) отвечает за проверку подлинности связи между субъектом сертификата и его ключом. В случае RPKI также удостоверяется, что субъект сертификата имеет права на использование номерных интернет-ресурсов (НИР), перечисленных в расширении. По существу, эта функция неотличима от функции регистрационных услуг, оказываемых сегодня РИР. Хотя предполагается, что УЦС удостоверяет субъекта CP, данное условие не является необходимым, и информация о субъекте в CP может не быть неявной (например, субъект может быть представлен цифровым идентификатором, имеющим значение только во внутренней структуре УЦС).

Репозитории — открытые базы данных, в которых публикуются выданные сертификаты, списки аннулированных сертификатов и в случае RPKI — вторичные объекты.

Сертификаты.
Система RPKI основана на сертификатах стандарта X.509. Они содержат критические расширения для документации АИР, стандартизованные в RFC 3779[18]. Такое расширение содержит список всех АИР (адресов IPv4 и IPv6, а также номера автономных систем), полученных субъектом сертификата. Важно

[17] RFC 6480: An Infrastructure to Support Secure Internet Routing, URL: https://www.rfc-editor.org/rfc/rfc6480

[18] RFC 3779: X.509 Extensions for IP Addresses and AS Identifiers, URL: https://www.rfc-editor.org/rfc/rfc3779

отметить, что в задачу CP не входит идентификация субъекта, в отличие от стандартной системы PKI. CP удостоверяет, что УЦС распределил определенные ресурсы субъекту сертификата и данные ресурсы перечислены в расширении сертификата. УЦС удостоверяет, что любой документ, подписанный секретным ключом, соответствующим открытому ключу сертификата, подписан законным обладателем прав на использование ИР, перечисленных в сертификате. Данная концепция схематично представлена на рис. 44.

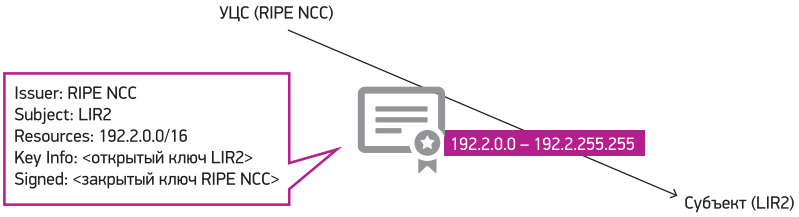

Рис. 44. X.509 Сертификат ресурсов с расширениями.

Вторичные объекты

Вторичные объекты специфичны для системы RPKI и, строго говоря, не являются ее частью, а предоставляются из соображений удобства практического применения RPKI. Они являются данными, подписанными владельцем сертификата. Один из наиболее проработанных вторичных объектов — так называемое разрешение на создание маршрута, или ROA (Route Origin Authorisation), определенное в RFC 6482[19]. Как следует из названия, ROA является разрешением, выданным сетью-владельцем прав на использование АИР.

Это разрешение на анонсирование данных ресурсов автономной системой (АС), указанной в ROA. В соответствии со спецификацией ROA содержит номер авторизованной АС и список IP-префиксов, которые эта АС имеет разрешение анонсировать. К этому «заявлению» прилагается сертификат, описывающий соответствующие АИР, и весь объект подписан с использованием ключа, указанного в сертификате. Также отметим, что наличие ROA не означает «согласие» авторизованной АС на то, что указанные префиксы непременно будут анонсированы данной автономной системой.

Как и любая система PKI, RPKI имеет иерархическую структуру с корневым сертификатом во главе. Поскольку для корневого сертификата не существует родительского УЦС, данный сертификат представляет собой самоподписанный корневой ключ. Организация, которой принадлежит корневой CP, является так называемой точкой доверия (Trust Anchor). Важно отметить, что, как и в любой системе PKI, вопрос доверия данной PKI остается за третьими лицами — пользователями системы.

[19] RFC 6482: A Profile for Route Origin Authorizations (ROAs), URL: https://www.rfc-editor.org/rfc/rfc6482

Корневой СР в качестве списка АИР охватывает все адресное пространство IPv4, IPv6 и автономных систем. С помощью этого сертификата могут быть сгенерированы сертификаты РИР в соответствии с фактически распределенным адресным пространством.

В свою очередь РИРы могут сертифицировать АИР, которые они распределяют локальным регистратурам (Local Internet Registry — LIR) или конечным пользователям, которые получают АИР непосредственно от РИР. Локальные регистратуры могут осуществлять последующее распределение, а также соответствующую сертификацию. Наконец, конечные пользователи — сети, фактически использующие адресное пространство, — также должны иметь возможность генерирования временных сертификатов для подписания вторичных объектов RPKI, например, ROA. Дело в том, что срок жизни этих объектов обычно короче, чем право на использование АИР, а их аннулирование реализуется путем отзыва сертификатов, использованных при создании соответствующих объектов. Во избежание аннулирования всех вторичных объектов и последующего их воссоздания для каждого объекта генерируется свой СР. Таким образом, аннулирование вторичных объектов может быть осуществлено независимо.

Важное следствие такого подхода — все участники RPKI являются органами выдачи сертификатов, УЦС, а значит, нуждаются в соответствующей защищенной инфраструктуре, организации процессов управления ключами и т.п.

Общая схема RPKI проиллюстрирована на рис. 45.

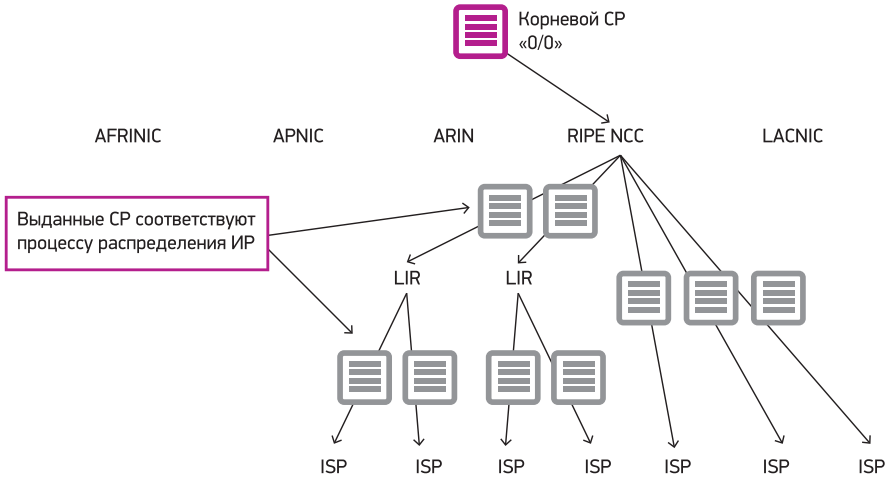

Рис. 45. Общая структура RPKI.

Использование RPKI

Одна из задач внедрения системы RPKI — предоставить возможность сетевым операторам независимо проверять достоверность сертификатов и ROA. Это важный процесс с точки зрения безопасности системы маршрутизации, и он состоит из нескольких этапов, которые мы кратко рассмотрим.

После проверки подлинности подписи ROA пользователь должен удостовериться, что ресурсы, описанные сертификатом ROA, содержат все IP-префиксы, указанные в ROA.

Следующим шагом является проверка так называемой цепи доверия, означающей, что начиная от корневого сертификата каждый последующий сертификат на пути к ROA выдан предыдущим и все сертификаты пути являются достоверными. Проверка достоверности в RPKI отличается от аналогичного процесса в PKI общего назначения: в RPKI, помимо проверки криптографической правильности сертификата, удостоверяется, что ресурсы, описанные родительским сертификатом, включают все ресурсы дочернего сертификата. Вы, наверное, заметили, что в отличие от системы PKI, используемой для веб-сайтов, валидация в RPKI производится, начиная от корневого сертификата, и предполагает проверку всего дерева. Это связано с громадным числом путей, которые требуется проверить: в случае 100-процентного внедрения системы RPKI число путей будет соответствовать числу маршрутов в глобальной таблице маршрутизации — и проверка снизу вверх просто непрактична.

Этот процесс описан в RFC 6487[20] и представлен на рис. 46.

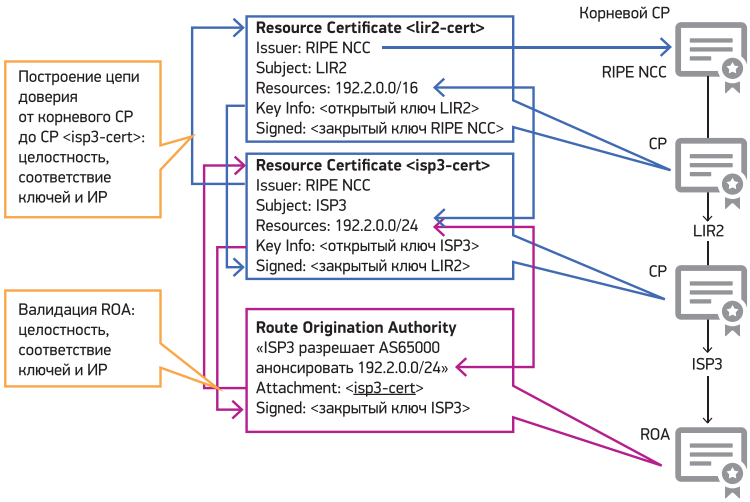

Рис. 46. Построение цепи доверия для проверки подлинности вторичных объектов и CP.

[20] RFC 6487: A Profile for X.509 PKIX Resource Certificates, URL: https://www.rfc-editor.org/rfc/rfc6487

Использование ROA возможно как для построения фильтров, так и в качестве дополнительного правила в процессе выбора пути BGP. Логично предположить, что интеграция в процесс BGP информации, полученной от системы RPKI, является более масштабируемым решением.

Архитектура такого решения схематично представлена на рис. 47. Предполагается, что сервис-провайдер хранит собственную копию всех объектов глобальной системы RPKI, проверяет их достоверность и периодически обновляет. Результирующая база данных содержит только достоверную информацию (достоверный кеш, validated cache) и может быть непосредственно использована процессом BGP маршрутизатора.

При получении очередного сообщения BGP маршрутизатор запрашивает базу данных на предмет наличия префиксов, указанных в поле NLRI сообщения BGP. Достоверность маршрута проверяется на предмет соответствия префиксов NLRI префиксам, указанным в ROA, а также номера первой AC пути (атрибут AS_PATH) номеру автономной системы, указанной в ROA как источник анонса. Если эти условия выполняются, маршрут маркируется как «valid». Если же база данных не содержит указанных префиксов, значит, система RPKI не содержит ни одного правомерного объекта ROA для этих префиксов. Причин может быть несколько. Например, просроченный сертификат в цепочке проверки подлинности ROA или отсутствие ROA как такового. Учитывая, что внедрение данной системы будет происходить постепенно, данная ситуация скорее свидетельствует, что сеть еще не использует преимущества RPKI, и такой маршрут может быть принят при отсутствии более надежного. В этом случае результатом определения достоверности маршрута будет «not-found».

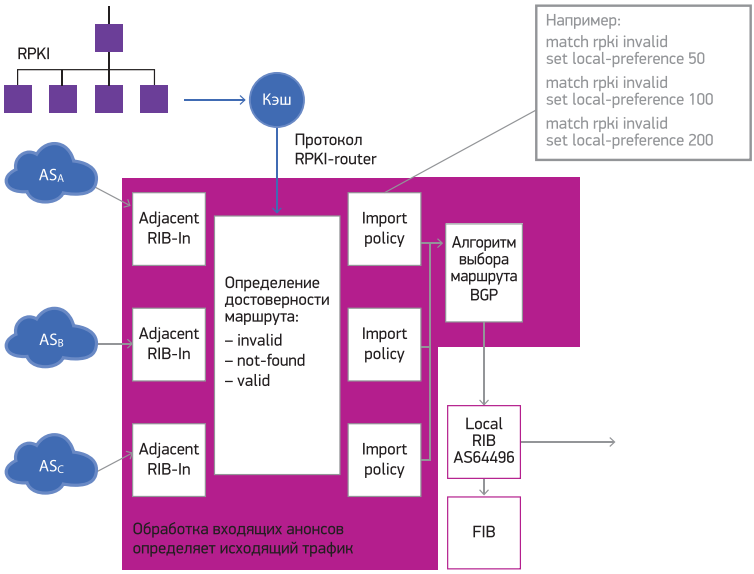

Рис. 47. Интеграция RPKI в процесс принятия решения BGP.

Другой случай — база данных содержит правомерные ROA, которые описывают префиксы, указанные в анонсе BGP, но не соответствующие автономным системам-отправителям, указанным в атрибуте AS_PATH. Возможно, это является свидетельством захвата префикса (prefix hijacking), и такой маршрут следует использовать с большой осторожностью. Скорее всего, его следует отбросить, даже если он единственный. Результат определения достоверности такого маршрута — «invalid».

Этот процесс определения достоверности маршрута получил название «валидация источника маршрута» (route origin validation, ROV).

В любом случае результат удостоверения маршрута с использованием RPKI является одним из критериев выбора маршрута в BGP наряду с другими атрибутами BGP — AS_PATH, ORIGIN, MED. В конечном счете интерпретация этой информации является частью политики маршрутизации данной сети. Например, на начальном этапе внедрения RPKI более разумной может быть политика занижения приоритета маршрутов «invalid» с помощью параметра LOCAL_PREF, как показано на рис. 45.

На сегодняшний день RPKI — наиболее технологичный способ обеспечения безопасности маршрутизации. Но даже если RPKI внедрят все операторы, это не обеспечит безопасность в полной мере. Возможность обмана в Сети останется, пока не будет поддержки функции установления подлинности пути передачи сообщения BGP — AS_PATH.

Например, злоумышленник может утверждать, что автономная система, указанная в ROA, является его клиентом: это реализуется путем присоединения номера данной АС в атрибут AS_PATH своих анонсов BGP. Поэтому RPKI не сможет полностью защитить глобальный Интернет от фальсификации адреса отправителя, атак типа Pilosov и YouTube. Но внедрение системы и, главное, применение сопутствующих технологий сервис-провайдерами позволят ограничить вред, наносимый такими атаками.

Поэтому в IETF были разработаны стандарты, обеспечивающие возможность проверки криптографической подлинности анонса маршрута и достоверности пути, по которому этот анонс был передан. Это расширение BGP получило название BGPsec.

Например, представим анонс маршрута 192.168/16 сетью AS_1 сети AS_2 и затем AS_3. По получении этого анонса сеть AS_4 сможет удостовериться, что AS_1 является правомочным «владельцем» маршрута 192.168/16, который она анонсировала сети AS_2; что сеть AS_2 получила этот маршрут от сети AS_1 и передала его сети AS_3; наконец, что сеть AS_3 получила этот маршрут от сети AS_2 и передала его сети AS_4.

BGPsec

Чтобы устранить описанные выше проблемы безопасности, требуется расширить возможности самого протокола BGP. Эти расширения были стандартизованы в 2017 году. Основной спецификацией протокола BGPsec является RFC 8205[21].

Расширение функциональности BGP реализовано с помощью нового атрибута: BGPSEC_Path_Signatures. Данный атрибут содержит последовательность цифровых подписей для каждой сети (точнее, автономной системы), через которую был передан соответствующий анонс маршрута.

Чтобы лучше понять, как это работает, представим три сети: AS_1, AS_2 и AS_3 (рис. 48). Допустим, что AS_1 анонсирует маршрут 192.168/16 сети AS_2. При использовании BGPsec этот анонс будет содержать атрибут BGPSEC_Path_Signatures, состоящий из префикса 192.168/16, сети-источника AS_1 и сети-пира, которой этот маршрут передан, — AS_2. Вся эта информация заверена подписью AS_1. В свою очередь, когда сеть AS_1 передаст этот анонс сети AS_2, та также заверит своей подписью информацию, полученную от AS_1, плюс номер автономной системы-пира, которой передается анонс, — AS_3. Заметим, что с точки зрения передачи анонсов от граничных маршрутизаторов требуется только наличие секретного ключа своей автономной системы для подписания атрибута BGPSEC_Path_Signatures.

Если AS_3 захочет удостовериться в подлинности полученного анонса, ей придется сделать несколько проверок. Сначала она должна будет убедиться, что AS_1 действительно авторизована для анонсирования этого маршрута держателем соответствующего адресного пространства. Для этого AS_3 проверит наличие и достоверность соответствующего объекта ROA системы. Затем эта сеть последовательно проверит подписи, содержащиеся в атрибуте BGPSEC_Path_Signatures полученного анонса, чтобы убедиться в их подлинности и соответствии пути прохождения анонса.

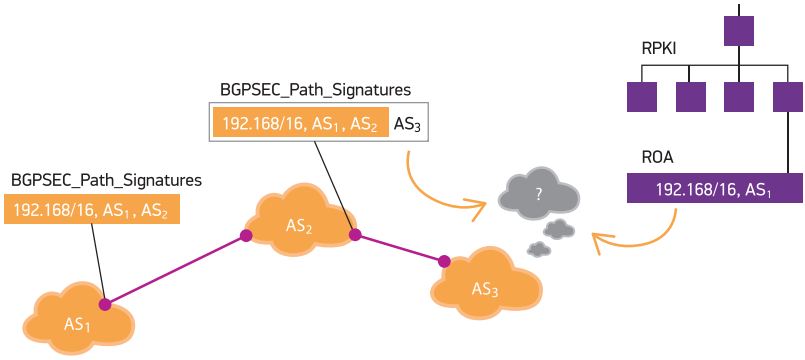

Рис. 48. Схема работы BGPsec.

[21] RFC 8205: BGPsec Protocol Specification, URL: https://www.rfc-editor.org/rfc/rfc8205

Чтобы обеспечить возможность проверки подписей граничных маршрутизаторов, BGPsec определяет дополнительный тип сертификата, связывающего открытый ключ маршрутизатора с номером его автономной системы. Этот сертификат является дочерним по отношению к сертификату RPKI данной АС. Таким образом, при верификации анонса проверяющая сторона сможет создать цепочку доверия, используя глобальную систему RPKI.

Как и в случае с RPKI, решение, которое примет сеть AS_3 по результату проверки, зависит от политики безопасности данной сети. Например, при наличии строгой политики рассматриваемый путь будет удален из таблицы маршрутизации, а более гибкая политика лишь уменьшит предпочтительность этого маршрута.

Пока работа над протоколом BGPsec находится на стадии стандартизации. Основная спецификация протокола «BGPsec Protocol Specification»[22] уже опубликована как стандарт RFC. Но пройдет немало времени, прежде чем этот стандарт и связанные с ним спецификации будут воплощены в оборудовании и затем внедрены сетевыми операторами. А пока операторы могут сфокусироваться на использовании результатов первой фазы работы — RPKI.

Более прагматичные решения

По мнению многих операторов, широкое внедрение BGPsec не является реалистичным в обозримом будущем. Производители сетевого оборудования тоже не спешат реализовать эту функциональность.

Во-первых, постепенное внедрение проблематично с экономической точки зрения, так как любая AS, которая оказывается на пути и не поддерживает BGPsec, сводит на нет все усилия и преимущества, делая валидацию невозможной. А таких AS на начальном этапе будет очень много. Во-вторых, BGPsec не защищает от другого вида атак – так называемых утечек маршрута (route leak).

Мы уже упоминали этот тип атак, давайте рассмотрим «утечку маршрута» более подробно. Суть этой атаки можно проиллюстрировать на простом примере. Допустим, сеть AS_1 (рис. 49) является клиентом двух провайдеров транзита – AS A и AS B. В случае, если этот клиент реанонсирует маршрут к Google Public DNS (8.8.8.0/24), полученный от AS B, другому провайдеру – сети AS A, а последняя примет его, то произойдет утечка маршрута. В результате весь трафик клиентов AS A к Google будет направлен через клиентскую сеть AS_1, вероятнее всего, с катастрофическими последствиями для последней.

[22] RFC 8205: BGPsec Protocol Specification,
URL: https://www.rfc-editor.org/rfc/rfc8205

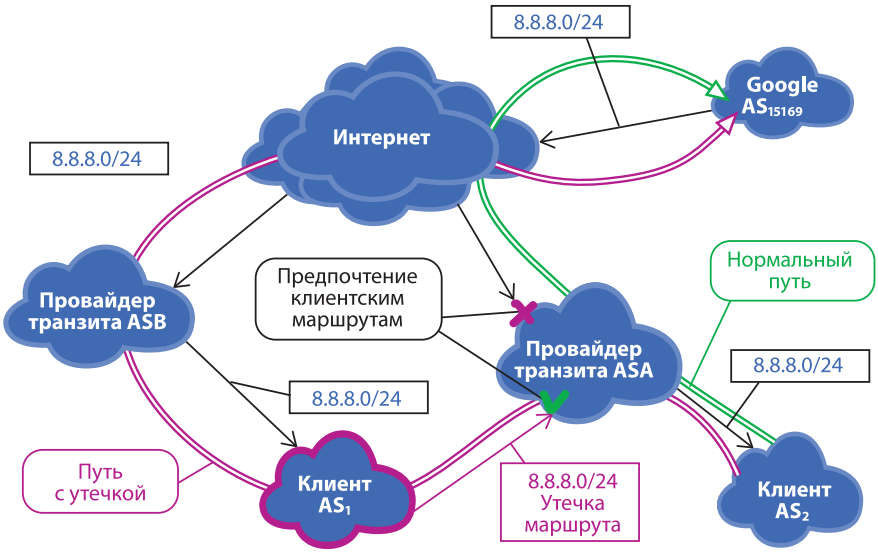

Рис. 49. Утечка маршрута сетью-клиентом.

Без дополнительных проверок со стороны сети AS A так и произойдет, поскольку типичная политика маршрутизации предполагает, что маршруты, полученные от клиентов, имеют предпочтение. При этом, конечно, политика предполагает, что клиент анонсирует только свои собственные сети, а не чужие сети где-то в Интернете, о которых он узнал от другого провайдера. Другими словами, в данном случае мы имеем дело с нарушением политики, а не манипуляцию пути. BGPsec в данной ситуации бессилен.

Но если на BGPsec рассчитывать не приходится, какие возможности улучшения защищенности маршрутизации есть в распоряжении сетевых операторов?

Peer-lock

Peer-lock (peer-locking) – это механизм, предложенный Джобом Снейдерсом (Job Snijders), в то время работавшим в NTT, обеспечивающий определенную защиту от утечек маршрутов, поступающих от пиров. В отличие от утечки, вызванной сетью-клиентом, как показано на рис. 47, утечки такого типа происходят в результате нарушения пиринговой политики – пиры должны анонсировать только собственные сети и сети своих клиентов. Если пир анонсирует сети, полученные от других пиров, это приводит к утечке маршрута.

Механизм Peer-lock основан на активной координации и требует кооперации от сети-пира, которая желает стать «защищенной» (“protected ASN”). В самом

простом варианте анонсы, включающие эту AS, позволены только через прямое пиринговое соединение. Анонсы, содержащие эту AS, но полученные от других пиров, будут отброшены.

Схема работы peer-lock показана на рисунке 50. Сеть NTT в этом примере является провайдером, обеспечивающим механизм. Защищенная сеть «peer A» указывает на возможность альтернативного транзита через сеть «peer B». В этом случае анонсы, содержащие B в пути будут приняты NTT только непосредственно от A или от B. Анонс с сетью B, полученный от сети C, будет отброшен. Таким образом, даже если сети A и C обмениваются маршрутами в режиме пиринга, peer-lock предотвратит возможную утечку маршрута через C.

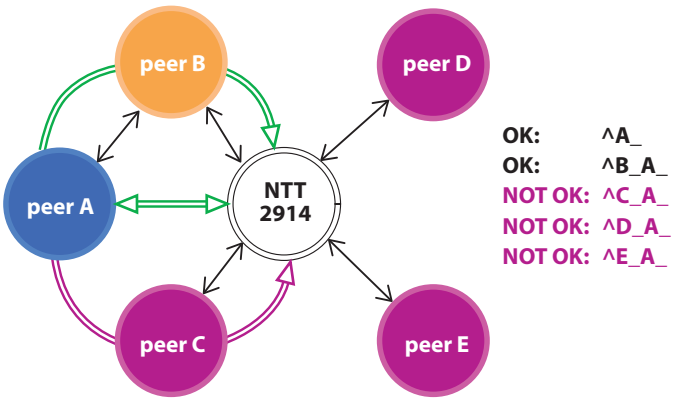

Рис. 50. Механизм работы peer-lock.
Источник: Job Snijders, https://archive.nanog.org/sites/default/files/Snijders_Everyday_Practical_Bgp.pdf

Механизм может быть адаптирован с учетом географической распределённости сетей[23].

Была также предпринята попытка расширить и автоматизировать механизм с использованием атрибута community BGP[24], но дальше изначального предложения дело не пошло. Некоторые из этих идей были использованы в другом предложении[25], работа над которым также не привела к конечному результату.

[23] https://instituut.net/~job/peerlock_manual.pdf
[24] https://datatracker.ietf.org/doc/draft-heitz-idr-route-leak-community
[25] https://datatracker.ietf.org/doc/draft-ietf-grow-route-leak-detection-mitigation

ASPA

ASPA, сокращенное от Autonomous System Provider Authorization, или Авторизация провайдеров автономной системы, позволяет автономной системе указать ее непосредственных провайдеров транзита. Подход основан на регистрации объектов ASPA, криптографически сертифицированных с помощью системы RPKI.

Согласно интернет-драфту «Verification of AS_PATH Using the Resource Certificate Public Key Infrastructure and Autonomous System Provider Authorization»[26], ASPA – это объект с цифровой подписью, который связывает для выбранного адресного пространства AFI (Address Family Identifier – идентификатор семейства адресов) (например, IPv4 или IPv6) набор номеров AS провайдеров с номером AS клиента (с точки зрения анонсов BGP, а не бизнеса) и подписывается владельцем AS клиента. ASPA подтверждает, что владелец клиентской AS (CAS) авторизовал набор провайдеров AS (Set of Provider ASes, SPAS) для дальнейшего анонсирования сетей клиента, например, провайдерам выше по потоку или пирам.

В этом смысле механизм похож на только что рассмотренный нами peer-lock. Однако peer-lock трудно автоматизировать, и информация о возможных транзитных провайдеров предназначена для конкретной сети, реализующей peer-lock, а не является частью глобального репозитория, как это предполагается в ASPA.

В отличие от BGPsec, который позволяет проверить всю цепочку пути анонса, ASPA позволяет определить валидность фрагментов пути. Процедура проверки сводится к нескольким шагам, описанным ниже. При этом предполагается, что проверяющая сторона, например, сеть, осуществляющая фильтрацию неверных анонсов, имеет доступ к кешу всех криптографически правильных объектов ASPA. Допустим, проверяющей стороне требуется проверить валидность фрагмента пути AS_PATH (AS_1, AS_2, AFI), а именно, предположение, что в случае, если AS_1 является клиентской AS, AS_2 является законным провайдером транзита для адресного пространства AFI.

1. Проверяющая сторона запрашивает из кеша все объекты ASPA, у которых CAS имеет значение AS1. Все SPAS, входящие в объединенное множество, являются потенциальными провайдерами.
2. В случае, если это множество пусто, проверка заканчивается с результатом «No Attestation» (Неизвестно).
3. Если AS2 является членом этого множества, проверка заканчивается с результатом «Provider+» (Верно).
4. В противном случае проверка заканчивается с результатом «Not Provider+» (Неверно).

При проверке полного пути все сегменты (hops) пути должны иметь значение «Provider+».

[26] https://datatracker.ietf.org/doc/draft-ietf-sidrops-aspa-verification/

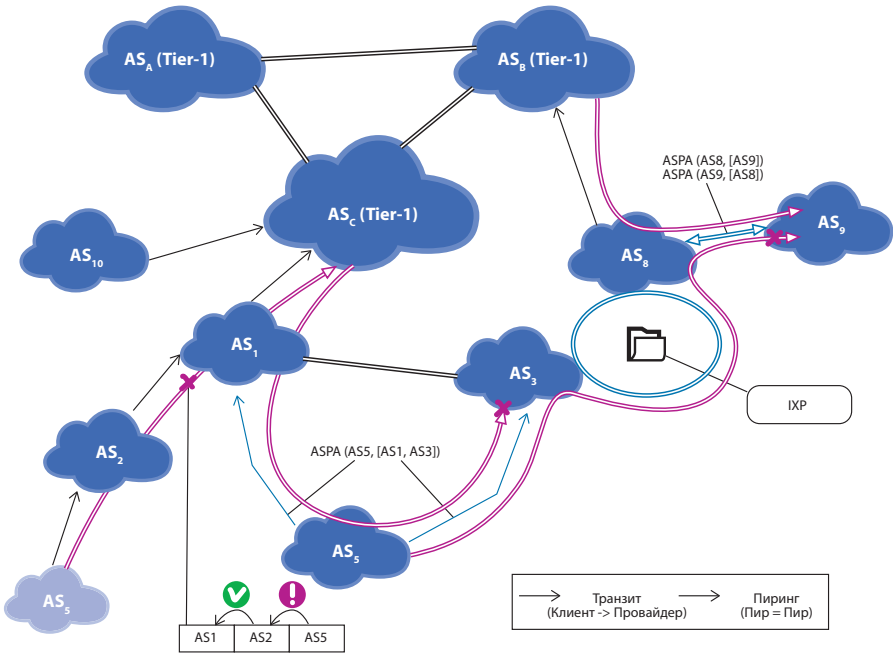

Рис. 51. ASPA позволяет предотвратить замаскированные захваты и утечки маршрута.

Как уже говорилось ранее, одной из задач защиты пути является предотвращение так называемых замаскированных захватов маршрута, когда атакующий представляется провайдером атакуемой системы. Так, на рис. 51 AS_2 может модифицировать AS_PATH анонсов сетей AS_5 провайдеру AS_1, добавив AS_5 в качестве клиента. Проверка с помощью ROA (ROV) в этом случае не поможет, поскольку формально AS_5 является источником анонсированных маршрутов. Хотя такой маршрут является менее «конкурентоспособным» в силу увеличенной длины, для многих сетей он может все же являться лучшим маршрутом, что приведет к успешной атаке.

Если же AS5 зарегистрирует своих транзитных провайдеров с помощью ASPA (AS_5, [AS_1, AS_3]), атакующему будет гораздо сложнее представиться провайдером AS_5 – фрагмент пути (AS_2, AS_5) будет отмечен как неверный в процессе проверки ASPA (см. рис. 51). Чем больше фрагментов пути будут зарегистрированы с помощью ASPA, тем меньше шансов у атакующего провести успешный захват маршрута.

Поскольку ASPA указывает на отношения между сетями (клиент-провайдер) и связанную с ними стандартную политику маршрутизации, этот механизм можно использовать для обнаружения утечек маршрутов. Так, AS_3 сможет установить, что AS_5 «протекла» с маршрутами, полученными от другого провайдера транзита AS_1, а AS_9 сможет остановить утечку маршрутов через последовательных пиров. Эти примеры приведены на рис. 51.

Важным свойством ASPA является то, что выгоды от его использования растут пропорционально масштабам внедрения этой технологии – в отличие от BGPsec, который требует значительного, если не тотального внедрения, прежде чем будет получен ощутимый эффект.

Роли участников сессии BGP

Как мы уже упомянули, атаки типа «утечки маршрута» являются результатом нарушения стандартной политики маршрутизации и потому защита BGP в этих случаях не работает.

Однако если BGP указать на тип взаимоотношений между сетями, участвующими в сессии, это может служить предохранителем против нарушения стандартной политики. Эта идея была стандартизована в IETF в 2022 году в спецификации RFC 9234[27].

Идея основана на том, что на начальном этапе при создании BGP-сессии между двумя сетями стороны должны договориться о возможностях, которые они обе поддерживают. К таким возможностям относится поддержка 4-октетных номеров автономных систем, BGPsec и т.п. Стандарт предлагает добавить еще одну возможность – определение роли каждой из сторон. В качестве таких ролей предложены следующие: Провайдер (Provider), Сервер маршрутов (Route Server, RS), Клиент сервера маршрутов (RS Client), Клиент (Customer), Пир (Peer).

С каждой ролью связана стандартная политика, так, например, стандартная политика в отношении Провайдера с Клиентом предусматривает, что Провайдер принимает все анонсы Клиента (и сетей его собственных Клиентов), а сам анонсирует глобальную связность (маршрут default или полную таблицу маршрутизации). Соответственно, Клиент зеркалирует эту политику.

Определение роли позволяет, во-первых, предотвратить ситуацию, когда стороны играют несовместимые роли (например, обе сети считают себя Провайдерами, типичный случай появления утечек маршрута), а во-вторых, предотвратить нарушение стандартных политик, соответствующих определенным ролям.

[27] RFC 9234: Route Leak Prevention and Detection Using Roles in UPDATE and OPEN Messages, URL: https://www.rfc-editor.org/rfc/rfc9234

Совместимыми являются следующие роли:

Местная АС	Удаленная АС
Provider	Customer
Customer	Provider
RS	RS-Client
RS-Client	RS
Peer	Peer

При обнаружении несоответствия ролей процесс создания BGP-сессии прерывается с ошибкой «Role Mismatch».

Стандарт также предлагает использование нового атрибута – «Only to Customer» или OTC. Этот атрибут представляет собой номер АС, который устанавливается в соответствии с процедурой, описанной ниже.

После того как роли сетей определены, этот атрибут может предотвратить анонсы, противоречащие стандартной политике маршрутизации для этой роли. Так, при получении маршрута от удаленной АС, сеть должна выполнить следующие проверки:

- Если маршрут с атрибутом OTC получен от Клиента или RS-Клиента, то это утечка маршрута, и такой анонс считается неприемлемым и должен быть отброшен.
- Если маршрут с атрибутом OTC получен от Пира, и атрибут имеет значение, не равное номеру АС этого Пира, то это утечка маршрута, и такой анонс считается неприемлемым и должен быть отброшен.
- Если маршрут получен от Провайдера, Пира или Сервера маршрутов RS, а атрибут OTC отсутствует, то он должен быть добавлен со значением, равным номеру удаленной АС.

Следующая процедура применяется к обработке атрибута OTC при анонсировании маршрута:

- Если маршрут должен быть анонсирован Клиенту, Пиру или RS-Клиенту (когда отправителем является RS), а атрибут OTC отсутствует, то при анонсировании маршрута атрибут OTC должен быть добавлен со значением, равным номеру местной АС.
- Если маршрут уже содержит атрибут OTC, он не должен анонсироваться Провайдерам, Пирам или Серверам маршрутов RS.

Нужно отметить, что, как и многие атрибуты BGP, OTC не защищен от модификаций (BGPsec защищает только атрибут AS_PATH). Поэтому описанный подход позволяет избежать ошибок конфигурации, но не умышленных атак. В то же время, по оценке специалистов, большинство инцидентов возникают как раз вследствие ошибок конфигурации, и в этом смысле роли и OTC являются эффективным средством.

Вопросы внедрения RPKI

Решения безопасности на базе RPKI зависят от уровня внедрения базовых компонентов – сертификации номерных ресурсов и создания ROA, соответствующих анонсам маршрутов.

Как видно из графиков на рисунках 52-55, наблюдается значительный рост числа ROA во всех регионах, как по количеству зарегистрированных префиксов, так и по размеру адресного пространства, которое они покрывают. По всем параметрам лидирует регистратура RIPE NCC, хотя ARIN не уступает по размеру адресного пространства IPv4 с зарегистрированными ROA.

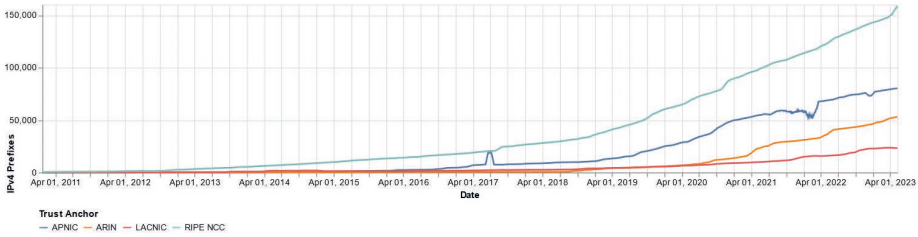

Рис. 52. Число префиксов IPv4 с зарегистрированными ROA.
Источник: https://certification-stats.ripe.net

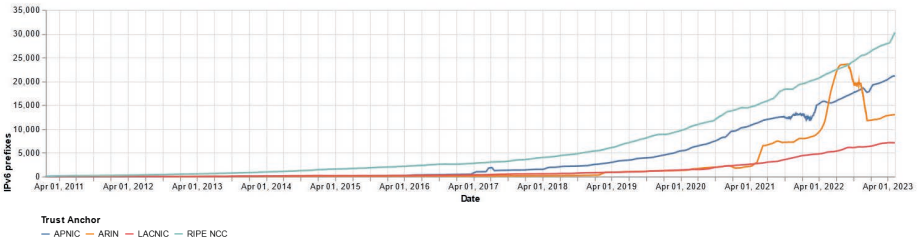

Рис. 53. Число префиксов IPv6 с зарегистрированными ROA.
Источник: https://certification-stats.ripe.net

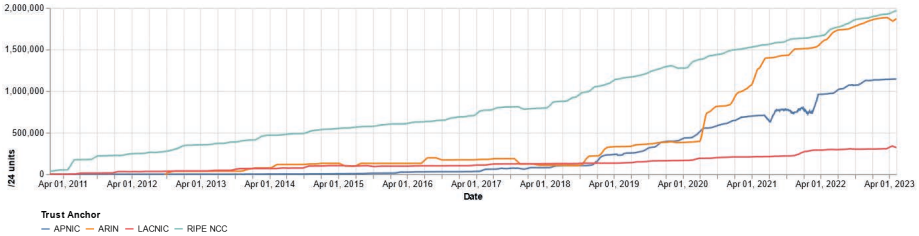

Рис. 54. Адресное пространство IPv4 (в единицах /24) с зарегистрированными ROA.
Источник: https://certification-stats.ripe.net

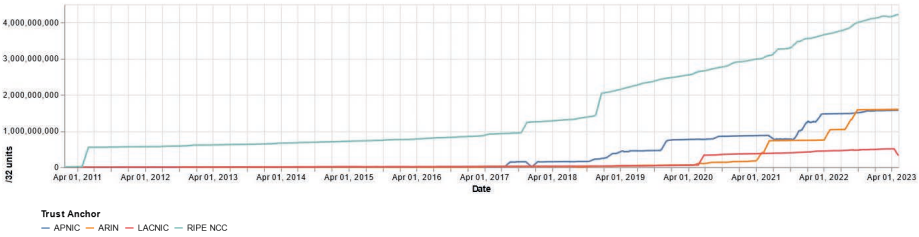

Рис. 55. Адресное пространство IPv6 (в единицах /24) с зарегистрированными ROA.
Источник: https://certification-stats.ripe.net

По состоянию на май 2023 года большая часть всех анонсов (67%) имела соответствующую ROA. Это означает, что 67% всех анонсов могут быть защищены с помощью RPKI в случае, если сети используют фильтрацию на основе этой технологии.

Здесь мы подходим ко второму параметру внедрения – уровню валидации источника маршрута (т.н. ROV) на основе сравнения полученных маршрутов с соответствующими ROA, о чем мы говорили в предыдущих разделах. Ведь для усиления защищенности системы маршрутизации требуются оба фактора – регистрация ROA владельцем адресного пространства и внедрение ROV операторами сетей.

Согласно исследованиям APNIC Labs, средний мировой уровень фильтрации на основе ROV не превышает 24% (по состоянию на май 2023 года). В этом смысле интересно сравнить карту внедрения ROA и использования ROV в мире.

Будем надеяться, что по мере более широкого покрытия анонсируемых префиксов ROA также будет расширяться использование этой технологии для фильтрации неверных маршрутов, делая общую систему более защищенной.

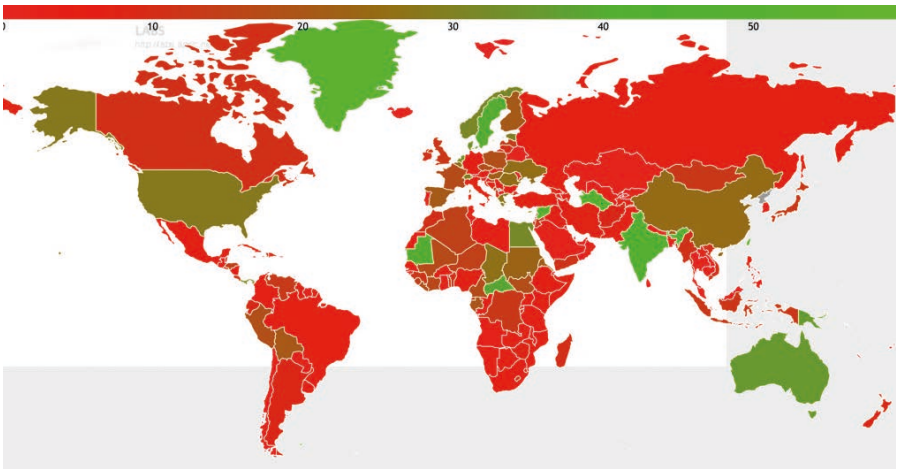

Рис. 56. Степень фильтрации маршрутов на основе ROV по странам.
Источник: https://stats.labs.apnic.net/rpki/

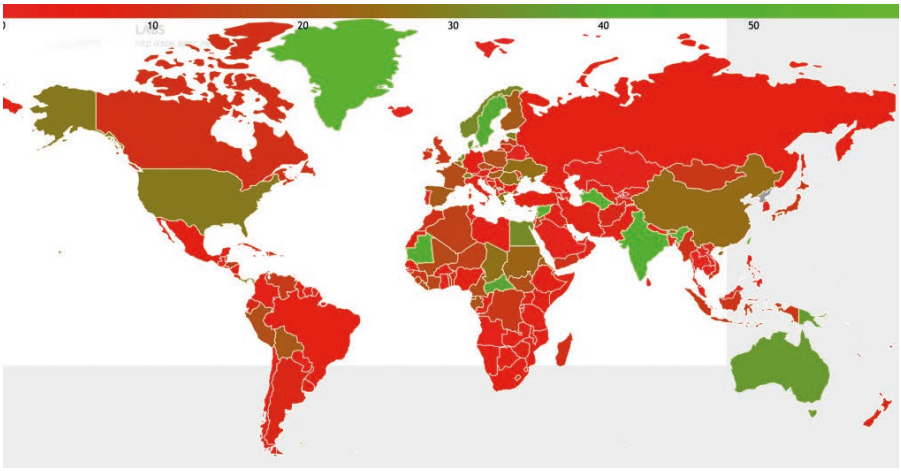

Рис. 57. Процент префиксов с зарегистрированными ROA по странам.
Источник: https://stats.labs.apnic.net/roa/

Вопросы обеспечения качества передачи в Интернете

Качество сеанса связи в Интернете непредсказуемо, но можно определенно говорить о его постоянном улучшении. Телеконференция Skype не уступает, а подчас и превосходит по качеству традиционную телефонную связь. Широкое распространение онлайн-игр и других интерактивных приложений говорит скорее о колоссальных возможностях Интернета, чем о его ограничениях. И все

же отсутствие базовой инфраструктуры, гарантирующей качество, подчас вселяет некоторый дискомфорт.

Во многом это ложное чувство. Хотя качество в Интернете и не обеспечивается на уровне базовой пакетной передачи, эта задача решается на многих уровнях, усиливая положительный эффект. Ключ к решению проблемы качества лежит в методах более эффективного использования имеющейся пропускной способности, о которых мы и поговорим в этом разделе.

Традиционная телефония и Интернет

Если взять стандартный спектр человеческого голоса и оцифровать его, используя простейший алгоритм импульсно-кодовой модуляции[28] (англ. PCM), то для передачи этой информации потребуется канал с пропускной способностью 64 кбит/с. Первые цифровые голосовые каналы обладали именно такой емкостью. Конечно, 64 кбит/с — это роскошь, и при использовании сегодняшних эффективных кодеков требуемую полосу можно значительно уменьшить. Например, кодек EFR (Enhanced Full Rate)[29], широко используемый в мобильной связи, позволяет «сжать» голос до 12,2 кбит/с. А один из базовых кодеков для приложений IP-телефонии, G.723.1[30], и вовсе использует полосу в 5,3 кбит/с!

В традиционной телефонии каждая сеть, через которую проходит вызов, должна сначала зарезервировать канал, а затем предоставить эту емкость для соединения или разговора, если вызываемый абонент снял трубку. В результате соединение между двумя говорящими имеет гарантированные пропускную способность и другие параметры качества, например, задержку. Задачей маршрутизации вызова и создания соединения в телефонии занимается система сигнализации ОКС-7 (англ. SS7)[31].

С другой стороны, если хотя бы одна из сетей перегружена и не может обеспечить канальную емкость, все предыдущие резервирования должны быть отыграны назад, и звонок не состоится. Однако достаточно простая топология и связность телефонных сетей, а также ограниченный спектр предлагаемых услуг — преимущественно голосовая связь или соединения одинаковой пропускной способности — позволяют обеспечить достаточно точное планирование емкостей и минимизировать число отказов.

Другой особенностью является то, что телефонные сети являются синхронными; по существу, они не используют буферизацию. Это означает, что задержка передачи между абонентами определяется в основном скоростью распространения сигнала или, другими словами, расстоянием между абонентами.

---

28 https://ru.wikipedia.org/wiki/Импульсно-кодовая_модуляция
29 http://ru.wikipedia.org/wiki/GSM-EFR
30 http://ru.wikipedia.org/wiki/G.723.1
31 https://ru.wikipedia.org/wiki/ОКС-7

Интернет же отличается коренным образом. Топология и связность сетей, как правило, весьма разветвленная и нерегулярная.

«Стандартной услуги» как таковой нет — различные приложения имеют различные требования к пропускной способности и качеству сети. Также Интернет является сетью пакетной передачи, где IP-дейтаграммы асинхронно передаются узлами-маршрутизаторами по каналам со значительной вариацией пропускной способности. Это, в свою очередь, выражается в нерегулярности трафика и требует использования буферов для сглаживания «всплесков». Как мы увидим дальше, буферизация и управление очередями пакетов являются важными факторами качества связи.

Качественные решения

Фундаментальные технологии и протоколы Интернета, такие как IP, TCP или BGP[32], были разработаны в соответствии с требованиями пакетной передачи данных и высокой стойкости в отношении отказа отдельных узлов или целых сетей. Эти технологии обеспечивали простую, но универсальную услугу — передачу данных в режиме «best effort», не гарантирующем никаких параметров качества, а иногда даже и того, что пакеты будут доставлены получателю. Сеть не давала предпочтения какому-либо приложению — все пакеты подвергались одинаковой обработке.

Предоставить такую услугу было относительно просто: не имели значения ни пропускная способность и производительность сети, ни инфраструктура нижнего уровня. Не требовалась синхронизация между отдельными сетями — основным критерием являлась поддержка протокола IP. Поэтому и «подключиться» к Интернету, стать частью сети сетей было так же просто — что и явилось катализатором его быстрого роста. Приложениям также не требовалось спрашивать никакого специального «разрешения» у Сети — то, что происходило выше уровня IP, было делом договоренности между отправителем и получателем, клиентом и сервером и т.п. Это и сегодня является основой громадного инновационного потенциала Интернета.

На первый взгляд, услуга best effort была довольно ограниченной — особенно во времена раннего Интернета, когда «традиционные» сети гарантировали достоверную доставку и высокие параметры качества передачи. Однако большинство тогдашних приложений Интернета были весьма «эластичными», а именно малочувствительными к параметрам передачи. Например, обмен файлами с помощью протокола FTP или отображение веб-страницы могло занять секунды, минуты или часы, но фундаментального значения для приложения это не имело.

Другое дело — приложения реального времени: голосовая связь и видеоконференции. Они требуют определенных параметров качества, ниже которых

[32] http://ru.wikipedia.org/wiki/TCP/IP

эти приложения работать просто не могут. Но для этих приложений существовали свои сети — традиционные телефонные и ISDN[33].

Однако дешевизна и растущее распространение Интернета заставило задуматься о возможности его использования также и для «качественных» приложений — видео и голоса. Это, в свою очередь, привело к разработке «качественных» решений.

IntServ, или Интеграция служб

В середине 90-х гг. прошлого века в IETF[34] была разработана концепция, получившая название Integrated Services, или IntServ (интегрированные службы). Как было указано в документе RFC 1633[35], описывавшем архитектуру IntServ, «это расширение [архитектуры Интернета] необходимо для удовлетворения растущих потребностей в услугах реального времени для широкого диапазона приложений, включая телеконференции, удаленные семинары и распределенное моделирование». Также IntServ должны были обеспечить мультиплексирование различных классов трафика в сети, что позволило бы операторам предоставлять различные изолированные друг от друга услуги, используя общую сетевую инфраструктуру.

IntServ определяет два основных класса приложений: приложения реального времени, чувствительные к задержке, и «эластичные» приложения, для которых не так важно, когда именно будут получены данные, а точнее, дейтаграммы.

Чувствительность к задержке у приложений реального времени также различается. Например, для передачи голосового или видеопотока приложению необходимо знать максимально возможную задержку для обеспечения адекватной буферизации. В противном случае поток будет прерываться и его качество деградирует. Для таких приложений IntServ предлагает так называемую гарантированную услугу, при которой задержка не может превысить заданную величину.

Менее чувствительные приложения смогут довольствоваться так называемым предсказуемыми услугами, дающими среднестатистическую задержку. Предполагается, что стоимость таких услуг будет значительно дешевле, так как этот подход позволяет достичь гораздо большей утилизации сети.

Наконец, «эластичные» приложения могут продолжать пользоваться «обычным» Интернетом с его услугой best effort.

---

33 http://ru.wikipedia.org/wiki/ISDN

34 http://www.ietf.org

35 RFC 1633: Integrated Services in the Internet Architecture: an Overview, URL: https://www.rfc-editor.org/rfc/rfc1633

Хотя сам подход казался привлекательным, он требовал двух существенных изменений в архитектуре и функционировании Интернета. Появилась необходимость требования контроля доступа и резервирования.

Действительно, для предоставления гарантий по задержке и пропускной способности сетевые ресурсы должны быть первоначально зарезервированы по всему пути передачи данных от источника к получателю (и обратно, поскольку большинство потоков являются дуплексными). Такой процесс очень напоминает резервирование канала в телефонии, только вариация параметров этих каналов значительно больше. Это, в свою очередь, потребовало внедрения дополнительного сигнального протокола резервирования (такой протокол был разработан — Resource ReSerVation Protocol (RSVP)[36]) и модификации протоколов внутри межсетевой маршрутизации.

Далее: прежде чем сеанс связи сможет состояться, приложение должно «заключить контракт» с сетью, в соответствии с которым сеть будет обеспечивать передачу данных. Этот «контракт» предусматривает, что приложение не выйдет за рамки установленных параметров, а сеть обеспечит входной контроль.

Схематично архитектура IntServ показана на рис. 58.

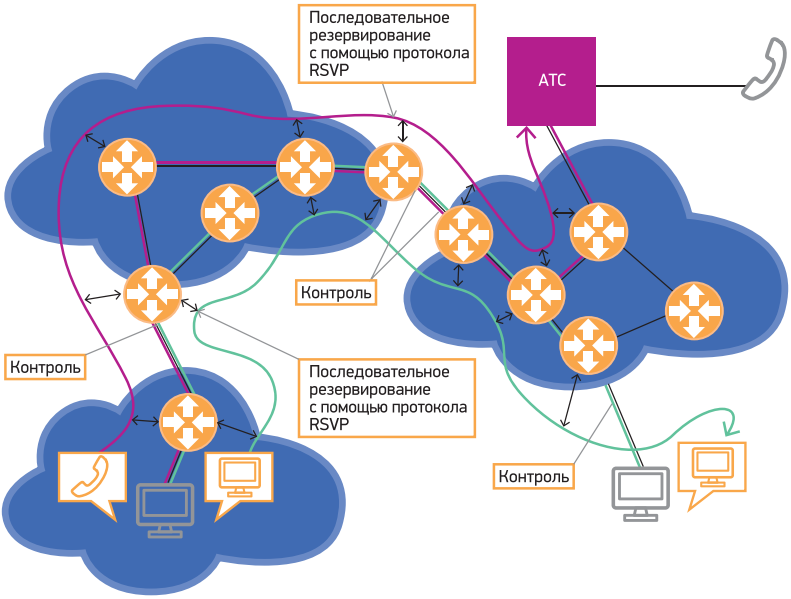

Рис. 58. Архитектура IntServ. На схеме представлено резервирование двух каналов с различными параметрами качества: для видеоконференции (зеленый цвет) и голосовой связи (красный цвет).

[36] RFC 2205: Resource ReSerVation Protocol (RSVP), URL: https://www.rfc-editor.org/rfc/rfc2205

Даже не вдаваясь в подробности, можно заметить, что уже на техническом уровне внедрение IntServ означало существенное усложнение архитектуры Сети. В экономическом смысле это влекло за собой существенное удорожание инфраструктуры для поддержки новой функциональности. Бизнес-отношения и система взаиморасчетов между сетевыми операторами должны были подвергнуться коренному пересмотру.

Наконец, технологии IntServ следовало внедрять и поддерживать глобально, во всем Интернете. Иначе — чего стоят островки качества в океане услуг «best effort»? Как и в случае многих других глобальных технологий, преимущества от их внедрения становятся ощутимыми, только когда они получают значительное распространение, — это своего рода замкнутый круг, который нелегко разорвать. Другими словами, даже одной из перечисленных проблем было достаточно, чтобы поставить под сомнение будущее предлагаемого решения. В случае с IntServ решение осталось в основном на бумаге.

DiffServ: разделяй и властвуй

Однако идея поддержки качества в глобальной инфраструктуре Интернета была слишком заманчивой, и IETF сделал вторую попытку. Началась разработка другого подхода, цель которого — обеспечение относительного, а не абсолютного, как в случае с IntServ, качества передачи. Другими словами, приложениям реального времени гарантируется определенная емкость, в рамках которой различные потоки конкурируют между собой. Эта архитектура была названа Differentiated Services, или DiffServ.

Вместо резервирования на уровне потока/приложения в DiffServ контроль производится на уровне достаточно статичных агрегированных «профилей» трафика. Классификация пакетов и их принадлежность к тому или иному профилю определяется по полю IP Type of Service (TOS), исторически оставшемуся в заголовке IP-пакета.

Соответственно, соглашение между различными сетями должно включать и договоренность о параметрах ограниченного числа профилей. При этом на входе в сеть производится контроль и возможное кондиционирование трафика для каждого профиля, которое включает буферизацию или даже отброс пакетов, если агрегированный поток превышает договоренные параметры.

Масштабируемость данного подхода, безусловно, выше по сравнению с IntServ. Еще важнее, что DifServ не требует динамического резервирования и сохранения состояния для каждого узла сети и каждого потока, проходящего через нее. Однако неразрешимым вопросом остается проблема предоставления определенного качества индивидуальному приложению. Трудно представить, что динамические требования многообразных приложений Интернета можно уложить в прокрустово ложе статической конфигурации нескольких профилей.

В результате и это решение не вызвало большого энтузиазма среди операторов. Ресурсы и инвестиции, требуемые для внедрения «качественных» решений, нашли лучшее применение в увеличении канальной емкости, благо и стоимость этих каналов к концу 1990-х значительно упала.

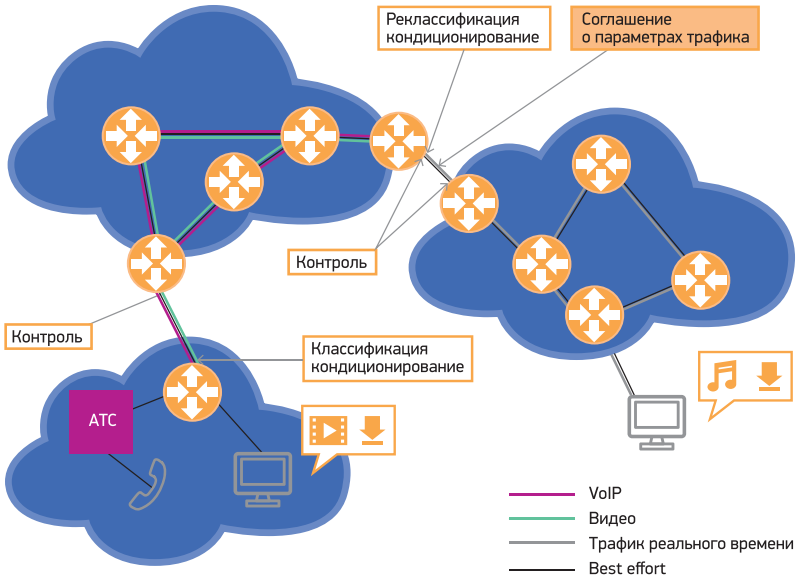

Рис. 59. Система DiffServ, позволяющая управлять качеством ограниченного числа «профилей» трафика. На схеме: голосовой трафик, видео и обычный трафик best effort.

Заметим, что концепция DiffServ используется для оптимизации трафика во внутренней инфраструктуре сети оператора, и здесь производители оборудования всегда рады помочь с широким набором возможностей, однако задача обеспечения сквозного гарантированного качества в Интернете по-прежнему остается мечтой. О полезных применениях DiffServ мы поговорим чуть позже.

Скрытые резервы

Колоссальный рост доступной канальной емкости и производительности сетей отодвинул проблему качества на задний план. Приложения реального времени также стали «умнее» и адаптивнее. Но по мере того, как опорные сети освобождались от заторов, узкое горлышко начало проявляться ближе к самому пользователю. И проблема оказалась в самом неожиданном месте — в неоправданно больших буферах устройств передачи, собственно и призванных бороться с перегрузками в сети. Феномен этот получил название bufferbloat, или «раздутый буфер».

Bufferbloat

Чтобы понять, в чем же тут дело, необходимо посмотреть, как работает фундаментальный протокол передачи данных в Интернете — транспортный протокол TCP[37].

Поскольку TCP — протокол достоверной доставки, то передача данных отправителем основана на получении подтверждения от получателя. Отсутствие подтверждения позволяет отправителю заключить, что произошла потеря пакетов, и произвести повторную передачу. Однако, если отправитель будет вынужден ожидать подтверждения каждого пакета перед передачей очередного, передача данных станет неэффективной и не сможет достичь максимальной скорости — ведь пока пакет достигнет получателя, а затем подтверждение проделает обратный путь, канал будет простаивать.

Современный дизайн TCP оптимизирован для максимальной производительности. Отправитель пытается заполнить канал данными, чтобы минимизировать простои. В то же время протокол должен эффективно работать и в случае возможной потери данных, когда требуется повторная передача.

Для этого в TCP используется понятие «окна», определяющего объем данных, которые могут быть переданы без получения подтверждения. Размер окна выбирается из следующих соображений. Для получения подтверждения на переданный пакет потребуется как минимум время для путешествия данных туда и обратно, так называемые round-trip time (RTT). Обозначим пропускную способность канала как BW (bandwidth), и тогда объем данных, определяемый произведением RTT x BW, целиком заполнит канал в интервале между подтверждениями. В этом случае отправитель будет передавать данные без остановки и канал будет до предела заполнен пакетами. Например, если RTT = 100 мс, а скорость передачи 100 Мбит/с, то окно равняется 10 Мбит. Это означает в идеальном случае, что в тот самый момент, когда отправитель передаст последний бит, будет получено подтверждение и передачу можно будет продолжать.

Проблема заключается в том, что в реальности отправитель не знает ни эффективной пропускной способности пути между ним и получателем, ни RTT. Для поиска правильных параметров в TCP существует режим так называемого медленного старта (slow start), когда изначальное окно задается размером один пакет, а затем экспоненциально растет с получением каждого последующего подтверждения. Это происходит до тех пор, пока не обнаруживается потеря пакетов, после чего отправитель уменьшает скорость передачи и переходит в режим «избежания перегрузки» (congestion avoidance).

Если мы взглянем на путь передачи данных между отправителем и получателем, то увидим, что, скорее всего, он проходит через несколько физических сетей,

[37] http://ru.wikipedia.org/wiki/TCP

каждая — со своей канальной емкостью и уровнем загрузки. Очевидно, что эффективная пропускная способность этого пути будет определяться участком с самой низкой пропускной способностью. Для сегодняшнего пользователя таким «узким горлышком» часто является домашний маршрутизатор или кабельный модем, где высокоскоростная беспроводная домашняя сеть (скажем, 54 Мбит/с) встречается с аплинком сервис-провайдера (например, 2 Мбит/с). Из-за разницы в скоростях пакеты, полученные из беспроводной сети, должны ожидать, пока очередной пакет будет передан через линк 2 Мбит/с. Для этого и используются буферы памяти. Буферы являются необходимым компонентом любой пакетной сети асинхронной передачи, они позволяют сгладить всплески трафика и улучшить общую производительность сети.

Но, к сожалению, это достигается не всегда. Наоборот: оказывается, буферы являются частью проблемы!

Логично предположить, что размер буфера должен быть сопоставим с размером максимального окна TCP — в этом случае буфер сможет удержать все пакеты максимального «всплеска», если таковой произойдет. Рекомендованный размер буфера = RTT × BW, где BW — пропускная способность исходящего канала (поскольку неизвестно, где находится истинное узкое горлышко), а в качестве RTT принято выбирать значение 100 мс — величину задержки трансатлантической передачи.

Удешевление компьютерной памяти и неполное понимание работы очередей привело к тому, что производители оборудования начали щедро начинять свои устройства буферами, иногда даже превышая рекомендованные размеры. И буферы стали частью канальной емкости, которую так хорошо умеет заполнять TCP!

И что же произошло? В случае появления затора буферы довольно быстро наполняются и переполняются, однако сигнал о потере (и, соответственно, о снижении скорости передачи) задерживается, поскольку пакет теперь проводит некоторое время в буфере, прежде чем продолжить свой путь к получателю и в виде подтверждения вернуться обратно. В качестве справки: для «очистки» буфера в 128 КБ при 3 Мбит/с аплинке (одна из типичных конфигураций в кабельных сетях, см. рис. 60) требуется 340 мс, в случае 1 Мбит/с аплинка — около 1 с.

В результате TCP не способен своевременно адаптировать скорость передачи к доступной канальной емкости, что вызывает задержки, потери и повторные передачи, способные на порядок уменьшить реальную пропускную способность.

Проблема усугубляется тем, что выбрать «правильный» размер буфера невозможно. Вариация параметров отдельных потоков данных, мультиплексируемых в канале, слишком велика и динамична для статически заданного

параметра. Более того, параметры самой физической среды могут меняться. Например, изменение местоположения ноутбука или планшета способно на порядок изменить пропускную способность в беспроводной сети. Вследствие этого и само «узкое горло» может меняться, перемещаясь от домашнего маршрутизатора в пользовательский компьютер.

Эффект bufferbloat может существенно снизить производительность сети, особенно для приложений, чувствительных к задержкам: ведь интерактивные приложения, VoIP и видео требуют, чтобы задержка не превышала 50–100 мс.

Данную проблему иллюстрируют данные, полученные с помощью приложения Netalyzr[38] для более чем 130 000 тестов, проведенных пользователями по всему миру. Каждой точке на графике соответствует отдельный тест. В каждом тесте скорость передачи постоянно увеличивалась, пока все буферы не оказывались заполненными. Из анализа распределения полученных пакетов исследователи делали вывод о пропускной способности канала (узкого горла) и возможном размере буфера[39]. Горизонтальные группы соответствуют наиболее типичным скоростям доступа, а вертикальные группы указывают на типичные размеры буферов. Наконец, диагональные линии показывают дополнительную задержку, порожденную буферизацией.

Как управлять очередями

В ряде случаев архитектура DiffServ может помочь решению проблемы. Поскольку DiffServ позволяет изолировать друг от друга потоки различных классов, то, например, для веб- и голосового трафика могут использоваться различные буферы. Но в то же время каждый из этих буферов подвержен проблеме bufferbloat.

Более основательное решение — использование алгоритмов активного управления очередями, или AQM (Active Queue Management). Первый из таких алгоритмов, RED[40], был разработан еще в 1993 году, когда проблема переполненных буферов впервые встала на повестку дня. Суть алгоритмов семейства RED заключается в постоянном мониторинге динамики уровня заполненности буфера и своевременной сигнализации различным TCP-потокам о необходимости снижения скорости путем маркировки или просто отброса пакета. К сожалению, уровень внедрения этих технологий остался невысок ввиду сложности настройки и негативного результата при неправильно выбранных параметрах (а во многих случаях и ввиду невозможности выбора оптимальных параметров для всех ситуаций).

---

38 http://netalyzr.icsi.berkeley.edu

39 Более подробно см. «Netalyzr: Illuminating The Edge Network», http://icir.org/ christian/publications/2010-imc-netalyzr.pdf

40 Random Early Detection, случайное раннее обнаружение, http://ru.wikipedia.org/ wiki/Random_early_detection

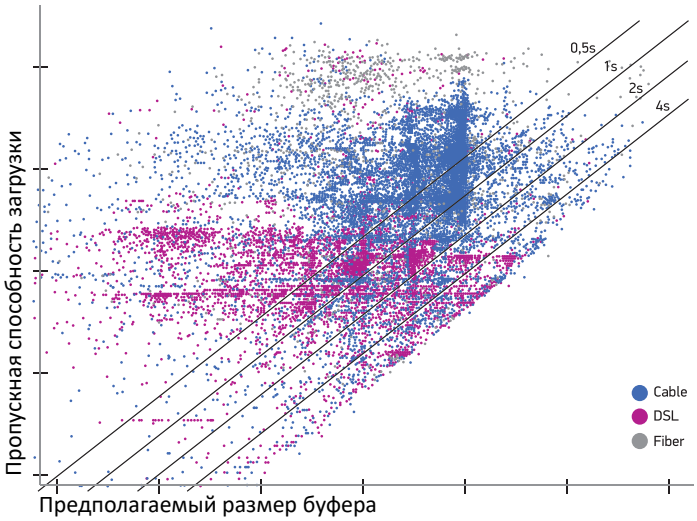

Рис. 60. Данные измерений пропускной полосы и возможных размеров буферов, полученные из тестов Netalyzr; вертикальные группы указывают на типичные размеры буферов в сети.
Источник: блог Jim Gettys (bufferbloat)
https://gettys.wordpress.com/2012/02/20/diagnosing-bufferbloat/

Контроль задержки: CoDel

Основываясь на опыте RED, Кати Николс (Kathie Nichols) и Ван Якобсон (Van Jacobson) предложили новый алгоритм. В отличие от предшественников, он не требовал ручной настройки. Алгоритм был назван CoDel от Controlled Delay, или контролируемая задержка. Ведь единственным целевым параметром является не размер очереди или ее статистические данные и не утилизация канала или степень отброса пакетов, а минимальная задержка, которой подвергаются пакеты в буфере.

Цель алгоритма CoDel — поддержание минимальной задержки ниже установленного параметра (5 мс). Если за определенный интервал, который изначально равен 100 мс, минимальная задержка превышает 5 мс, то последний пакет, покинувший очередь, отбрасывается, а интервал уменьшается в соответствии с формулой, отражающей число последовательных отбросов пакетов[41]. Как только минимальная задержка опять достигает заданного значения, отбрасывание пакетов прекращается и интервал принимает первоначальное значение в 100 мс.

Тестирование алгоритма показало хорошие результаты. ColDel позволяет удерживать задержку в районе заданного параметра для широкого спектра скоростей — от 3 до 100 Мбит/с. При этом утилизация канала составляет почти 100%.

[41] Более подробно см. Controlling Queue Delay, http://dl.acm.org/citation.cfm?id=2209336

Внедрение CoDel только начинается. В настоящее время этот алгоритм доступен для ОС Linux и OpenWrt[42]. К сожалению, в реальной жизни скорость внедрения соответствует циклу обновления пользовательского оконечного оборудования, который может составлять четыре–пять лет, а иногда и больше. Многое из сегодняшнего оконечного оборудования серьезно устарело и не позволяет внедрить новую функциональность путем простого апгрейда программного обеспечения. Тем не менее, хотя рассчитывать на быстрое решение проблемы не приходится, операторы должны иметь эту ситуацию в виду и не упустить очередного шанса улучшить качество предоставляемых услуг.

Управление потоками

Сразу оговоримся, что перегрузки в сети и связанные с ними заторы — неизбежное следствие вариации канальных емкостей по пути следования трафика и принципа работы протокола TCP, который старается максимально «заполнить» канал. Поэтому даже увеличение канальных емкостей не решает проблемы полностью. И если механизмы AQM призваны минимизировать излишнюю буферизацию в сети, тем самым уменьшая задержки и улучшая обратную связь, необходимую для контроля потоками, то новые разработки позволяют улучшить само управление потоками.

И здесь, как ни странно, может существенно помочь архитектура DiffServ, о которой мы уже говорили. Однако для этого требуется дополнительный инструментарий и другой взгляд на классификацию трафика.

В изначальной концепции DiffServ предполагалось, что трафик классифицируется в зависимости от приложения, которое его генерирует. Как мы обсуждали, небольшой набор стандартных профилей вряд ли обеспечит долгосрочное решение проблемы качества. Однако если мы посмотрим на совместное использование ресурсов сети не с точки зрения «бюджета» пропускной способности, а с точки зрения «бюджета» заторов, которые пользователь или приложение вызвали в сети, то ситуация станет намного проще.

Такой подход был применен крупным оператором широкополосного доступа в США — Comcast. Собственно, новая система была внедрена в ответ на претензии со стороны пользователей и регулятора (Федерального агентства по связи США) в отношении старой системы, которая для борьбы с перегрузками в сети осуществляла фильтрацию избыточного P2P-трафика, в частности, BitTorrent.

Новая система, внедренная в 2008 году и подробно описанная в RFC 6057[43], классифицирует трафик пользователя в зависимости от его вклада в перегрузку в сети (когда таковая случается). Работает система следующим образом:

---

42 https://openwrt.org

43 RFC 6057: Comcast's Protocol-Agnostic Congestion Management System, URL: https://www.rfc-editor.org/rfc/rfc6057

- по достижении определенного уровня загрузки сети (точнее — 70–80%) выявляются пользователи, на индивидуальную долю которых приходятся непропорционально высокие объемы трафика. Объем трафика считается высоким, если за последние 15 минут трафик пользователя превысил 70% предлагаемой полосы;
- трафик таких пользователей маркируется как трафик низкого приоритета. В маркетинговых терминах Comcast это означает изменение классификации от PBE (Priority Best Effort) на просто BE (Best Effort). Это не значит, что такой трафик обязательно будет отброшен. Однако, если сеть будет бороться с перегрузкой, применяя тот или иной алгоритм обработки очередей, пакеты BE будут обработаны в последнюю очередь (и если степень затора велика, то, возможно, частично отброшены);
- при последующем измерении маркировка трафика пользователя возвращается к PBE, если уровень его трафика опустился ниже 50%.

Данный подход, вкупе с оптимальной буферизацией и обработкой очередей, о которых мы говорили выше, хорошо зарекомендовал себя на практике. При его применении «штрафуются» не отдельные приложения, а пользователи, перегружающие сеть на протяжении длительного времени. При этом в условиях низкой загрузки сети доступна вся ее пропускная способность.

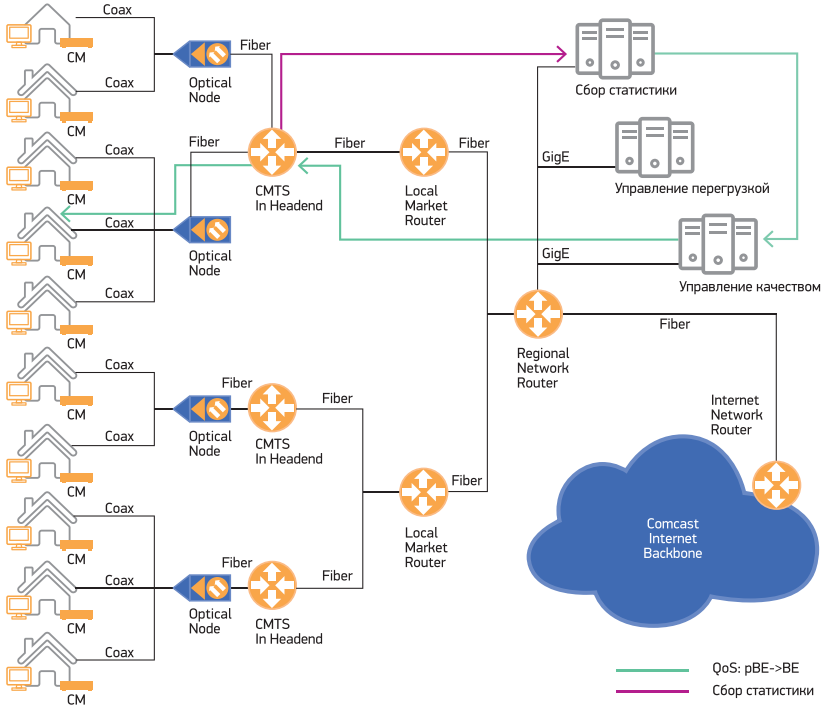

Рис. 61. Упрощенная схема сети широкополосного доступа компании Comcast и системы управления заторами.
Источник: http://downloads.comcast.net/docs/Attachment_B_Future_Practices.pdf

ECN и Conex

RED и CoDel позволяют оптимизировать буферизацию пакетов и тем самым увеличить эффективную пропускную способность пути передачи данных. Однако естественным способом управления потоком все же является отбрасывание пакета, поскольку это сигнал отправителю данных о перегрузке и о необходимости снизить скорость передачи. С другой стороны, отбрасывание пакета — это потеря данных, требующая повторной передачи и, соответственно, приводящая к снижению эффективной пропускной способности.

Поэтому и встает вопрос: нельзя ли сигнализировать о перегрузке каким-либо другим способом, без потери данных?

Для этого в IETF был стандартизован механизм под названием Explicit Congestion Notification[44] (ECN, или «явное уведомление о перегрузке»). ECN основан на маркировке пакетов специальным флагом вместо отбрасывания пакета в режиме перегрузки. Маркировка пакетов происходит на уровне IP путем установки соответствующих битов в IP-заголовке пакета, ведь перегрузка обслуживается маршрутизаторами именно на этом уровне.

Применение ECN не является обязательным. Но отправитель и получатель могут договориться о его использовании в момент установления TCP-соединения. В этом случае все пакеты отправителя для данного соединения будут содержать бит ECT (ECN Capable Transport) в IP-заголовке. В случае перегрузки в сети для пакетов ECN промежуточные маршрутизаторы вместо отбрасывания пакета могут установить флаг CE (Congestion Encountered).

Получатель, в свою очередь, должен отправить уведомление о перегрузке обратно отправителю, но в данном случае — в TCP-заголовке сегмента. При получении такого уведомления отправитель снизит скорость передачи точно так же, как он сделал бы в случае обнаружения потери пакета.

Участники рабочей группы IETF Conex[45] пошли еще дальше, поставив перед собой задачу сделать информацию о сквозной перегрузке всего пути видимой не только отправителю и получателю, но и сетям, через которые этот трафик проходит.

Согласно Conex, отправитель данных указывает уровень перегрузки для данного потока по всему пути, основываясь на информации от получателя. А получатель определяет уровень перегрузки на основе полученных пакетов с маркировкой ECN (CE), а также потерянных пакетов. Хотя эта информация относится к недавнему прошлому (поскольку поступает с задержкой в RTT), это все же лучше, чем ничего.

---

44 RFC 3168: The Addition of Explicit Congestion Notification (ECN) to IP, URL: https://www.rfc-editor.org/rfc/rfc3168

45 Congestion Exposure, http://datatracker.ietf.org/wg/conex/charter

Теперь любой маршрутизатор по пути следования трафика может определить не только уровень перегрузки на собственном участке, но и на других отрезках пути — как вверх, так и вниз по потоку. Для определения перегрузки вверх по потоку маршрутизатору достаточно взглянуть на уровень пакетов с маркировкой CE/ECN, а если вычесть это значение из предполагаемой перегрузки для всего пути (как указано отправителем), то получится уровень перегрузки по потоку вниз.

Эта информация имеет различное применение. Например, на ней может быть основана система управления потоками Comcast, использующая более точные данные о фактической перегрузке, вызванной тем или иным потоком. И тогда кондиционированию будет подвергаться не весь трафик пользователя, а отдельные потоки, вносящие существенный вклад в перегрузку. Другой пример применения — получение оператором детальной информации о текущей загрузке различных участков сети.

Топологическая эволюция

Многие из описанных технологий и подходов позволяют оптимизировать загрузку сети, порой высвобождая значительные ресурсы. Сквозная оптимизация, однако, гораздо более сложная задача, поскольку требует дополнительной кооперации между независимыми сетями и усложняет сетевое взаимодействие большим числом параметров. Поэтому сквозные решения имеют ограниченное применение.

Реальному прорыву в обеспечении качества передачи и сквозной производительности мы должны в значительной степени быть благодарными топологической эволюции Интернета, а именно появлению и развитию сетей доставки контента (content delivery networks, CDNs). Но об этом мы поговорим в главе 5 «Будущее начинается сегодня».

Эволюция системы маршрутизации: программируемый Интернет

Интернет все глубже проникает в нашу жизнь и продолжает удивлять новыми возможностями. Но в основном развитие сегодняшнего Интернета происходит на уровне приложений. Каждый день приносит нам сотни, если не тысячи новых apps. Спускаясь ниже, к сетевому уровню, мы обнаружим, что здесь развитие происходит существенно медленнее. Безусловно, «количественно» базовая инфраструктура Интернета продолжает стремительно развиваться — растет общая пропускная способность, внедряются новые технологии канального уровня. Но архитектурно изменения до последнего времени были весьма незначительны. В частности, потому, что сетевая архитектура представляет собой достаточно монолитный блок, включающий множество функций, в том числе и сетевые «приложения», как, например, DNS или BGP. Внедрение новой функ-

циональности требует модернизации всего сетевого стека в миллионе устройств. Представьте, что вам пришлось бы делать апгрейд операционной системы компьютера каждый раз, когда вы устанавливаете новое приложение!

Другими словами, инновация глобальных инфраструктурных протоколов и приложений в рамках сегодняшней архитектуры весьма затруднительна. Новые функции и возможности увеличивают сложность системы, их тестирование трудоемко, а внедрение сложно реализуемо, поскольку требует глобального масштаба. Мы обсудим эти вопросы подробнее в главе 4 «Экосистема Интернета». Не обошли эти проблемы и систему маршрутизации, и используемые протоколы, такие как BGP и OSPF. В рамках существующей парадигмы, когда каждый узел принимает независимое решение на основе информации, полученной от соседей, очень трудно решать глобальные задачи оптимизации трафика и реализации соответствующей политики маршрутизации.

Поэтому многие связывают большие надежды с новой сетевой моделью, получившей название SDN (Software Defined Networking), или программно-конфигурируемая сеть. В чем же новизна подхода, и действительно ли это новое архитектурное знамение?

Пакеты, коммутация и маршрутизация

Чтобы лучше представить архитектурный сдвиг, предлагаемый SDN, вернемся к общим вопросам маршрутизации в IP-сетях.

Различают внутреннюю (по отношению к сети) и внешнюю, или межсетевую, маршрутизацию. Эти две системы используют различные протоколы. Например, для внутренней маршрутизации широко используется протокол OSPF[46], а стандартом межсетевой маршрутизации, как мы говорили, является BGP. Ключевым элементом сетевой инфраструктуры является маршрутизатор, который выполняет две функции: собственно маршрутизацию — определение лучшего маршрута к получателю данных — и пересылку пакетов с одного интерфейса на другой. Иногда под маршрутизацией понимают обе функции, но на деле они существенно различаются.

В алгоритмическом плане пересылка пакетов достаточно проста, и основной задачей является производительность. Отправителями и получателями пакетов являются интерфейсы маршрутизатора, а определение адресата осуществляется с помощью так называемой таблицы передачи — Forwarding Information Base (FIB).

Задачей же маршрутизации является построение собственной таблицы — таблицы маршрутизации (Routing Information Base, RIB), которая потом транслируется в таблицу FIB. Для построения этой таблицы и используются протоколы маршрутизации, которые на основе информации, полученной от соседних

[46] http://ru.wikipedia.org/wiki/OSPF

маршрутизаторов (например, о связности и доступности тех или иных маршрутов), и собственной конфигурации (например, статических маршрутов и ограничений, наложенных сетевой политикой маршрутизации) формируют собственное представление о сети и ее топологии.

Важным в этой модели является то, что каждый маршрутизатор принимает решения самостоятельно и относительно независимо.

Такая модель работает замечательно в простых сетях, особенно когда основной задачей является обеспечение связности. Она проста и надежна. Однако по мере усложнения сетевой архитектуры и политики внутренней и внешней маршрутизации ограничения модели становятся все более заметными.

Возьмем, к примеру, необходимость включения в политику маршрутизации требований обеспечения определенных параметров качества, производительности и стоимости. Эта задача зачастую невыполнима из-за ограничений существующих протоколов маршрутизации. А попытки удовлетворить такие требования в рамках протоколов уже привели к их значительному усложнению и созданию закрытых расширений.

Аналогичные проблемы присутствуют и в сетях Ethernet или MPLS, когда автономность элементов и связанная с этим необходимость распределенной конфигурации существенно усложняют управление сетью.

Программно-конфигурируемая сеть

Концепция SDN позволяет разрабатывать и конфигурировать сети при помощи программирования.

В традиционной архитектуре в одном устройстве сосуществуют уровень управления (так называемый control plane, к которому относятся, например, процессы маршрутизации) и уровень передачи данных (так называемый data plane, отвечающий за пересылку пакетов с одного интерфейса на другой). Концепция SDN предусматривает передачу управляющих функций центральному серверу — так называемому контроллеру, таким образом заменяя традиционную распределенную модель маршрутизации на централизованную. Соответственно, и процесс управления сетью, включающий создание маршрутов, является не чем иным, как программированием сети в целом. Для сравнения на рис. 6о приведены традиционная сетевая архитектура и сеть SDN.

Такой подход обладает рядом существенных преимуществ.

Во-первых, существенно упрощается сам процесс создания маршрутов. В сегодняшней сети маршрутизация — это распределенный итеративный процесс, при котором рабочая топология сети «вычисляется» совместно всеми устройствами. А в SDN это не что иное, как программа моделирования сети с заданными параметрами. Использование этой модели открывает новые воз-

можности по созданию сети с требованиями, немыслимыми в рамках традиционного инжиниринга трафика и с использованием стандартных протоколов маршрутизации.

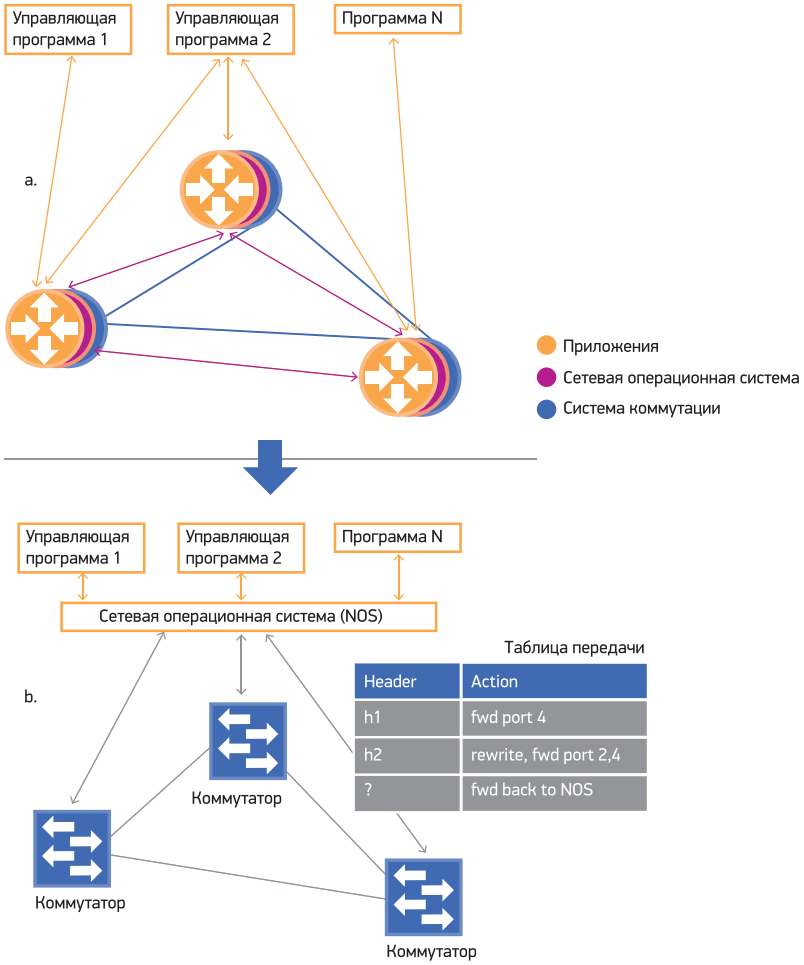

Рис. 62. Разделение системы управления и передачи в архитектуре SDN: a — традиционная архитектура с автономными сетевыми элементами; b — архитектура SDN с централизованной управляющей функцией).

Рассмотрим в качестве примера реконфигурацию сети в случае падения канала между какими-либо узлами сети. В традиционной модели маршрутизатор, подключенный к данному каналу, сообщит своим соседям о разрыве связи — и все они независимо займутся разработкой новых маршрутов. При этом они будут распространять информацию об изменяющейся топологии своим соседям и т.д.

В конце концов итерационный процесс закончится и сеть перейдет в новое состояние. В зависимости от сложности сети и используемых протоколов маршрутизации, этот процесс займет больше или меньше времени, но, учитывая задержки при передаче информации на каждой итерации, всё произойдет совсем не мгновенно. Более того, этот процесс не является детерминированным: другими словами, если повторится падение того же канала — не факт, что сеть перейдет в то же состояние.

В случае использования управляющего центра расчет новой топологии производится исходя из знания обо всей сети в целом. Мы также можем задать необходимую топологию следующего состояния. Наконец, поскольку создание новой топологии — это чисто вычислительный процесс, он может быть выполнен значительно быстрее. Во-вторых, значительно увеличиваются возможности для инноваций. В традиционной распределенной модели необходимо привести функциональность к общему знаменателю — для возможности взаимодействия между независимыми устройствами. Это определяет существенную консервативность системы по отношению к новшествам и приводит к «технологической окостенелости», что мы во многом наблюдаем в сегодняшней глобальной инфраструктуре Интернета. В SDN же инновация — лишь вопрос написания нового приложения. Наконец, в-третьих, вместо сложных и дорогостоящих маршрутизаторов можно использовать более простые устройства.

OpenFlow

SDN и OpenFlow — не одно и то же. SDN означает более общую архитектурную концепцию отделения уровня передачи данных от уровня управления. OpenFlow — протокол, наиболее проработанный и стандартизованный, который реализует эту концепцию. Далее мы также рассмотрим несколько другой подход к реализации SDN, основанный на программировании сети с использованием существующих протоколов маршрутизации.

Разработка OpenFlow началась с исследовательского проекта, который должен был дать возможность разработчикам новых сетевых архитектур и протоколов опробовать свои идеи в более или менее реальной среде — скажем, в рамках университетской сети, причем таким образом, чтобы внедрять новую архитектуру изолированно от существующих услуг и параметров сети, исключив негативное влияние на ее текущую функциональность.

Идея OpenFlow проста и основана на наблюдении, что, несмотря на существенные различия между сотнями моделей коммутаторов и маршрутизаторов, все они содержат таблицу передачи. А она определяет базовую функцию передачи данных — как можно быстрее переправить каждый входящий пакет на определенный исходящий интерфейс. Более того, хотя формат этих таблиц различен, можно идентифицировать стандартный набор функций данного уровня.

Каждая запись абстрактной таблицы передачи OpenFlow является «правилом» и связана с так называемым потоком данных (flow). Поток определяется заголовком

пакета — например, комбинацией адресов MAC, IP и номеров портов источника и получателя данных, хотя в принципе поток может состоять из пакетов с нестандартным заголовком — например, для поддержки внедрения новых протоколов. Не все элементы этой комбинации должны быть определены — например, поток может быть определен как весь трафик к некоторому хосту. В этом случае определенным является только один элемент — IP-адрес получателя данных.

Другим элементом записи таблицы является «действие» (action), которое определяет требуемую обработку пакетов потока. Основных действий четыре:

1. Передать пакет на определенный порт (или определенные порты) коммутатора.
2. Передать пакет контроллеру через «защищенный» канал. Контроллер — это управляющий центр сети, включающий центральную сетевую операционную систему и управляющие приложения, рассчитывающий топологию и маршруты, а также осуществляющий другие функции управления. Поэтому, как правило, первый пакет неизвестного потока отправляется контроллеру для определения правила и создания новой записи таблицы передачи.
3. Отбросить пакет. Это действие может быть необходимым, например, в борьбе с компьютерными атаками.
4. Пакет может быть направлен на «стандартную» обработку — например, если OpenFlow-коммутатор также является стандартным коммутатором или маршрутизатором. Данная функциональность позволяет отделить друг от друга потоки данных, управляемые OpenFlow, и потоки, управляемые другими механизмами, например, с помощью существующих протоколов маршрутизации. Благодаря этому исследователи могут изолировать экспериментальный трафик от нормального, используя общую инфраструктуру.

Наконец, последний элемент записи таблицы Open Flow содержит различную статистику — продолжительность потока, число полученных и переданных пакетов и т.п.

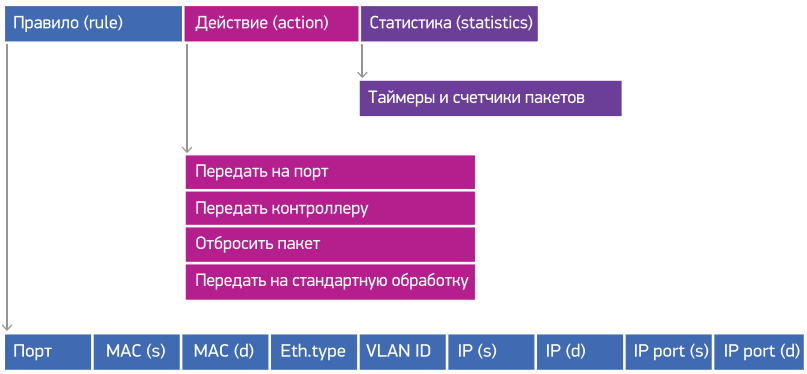

Рис. 63. Упрощенная структура таблицы передачи, OpenFlow 1.0.

На деле сегодняшняя архитектура OpenFlow[47] немного сложнее. В частности, она предусматривает наличие нескольких таблиц правил с возможностью каскадирования (см. рис. 64).

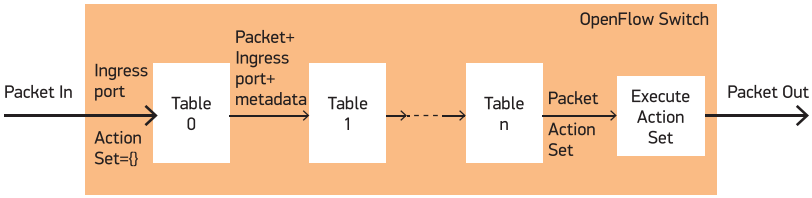

Рис. 64. «Каскадирование» правил и действий позволяет сделать обработку более гибкой.
Источник: спецификация коммутатора OpenFlow – OpenFlow Switch (Specification 1.5.1., https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.1.pdf)

Такая модель открывает уникальные возможности. Коммутатор OpenFlow может быть чем угодно — от простого коммутатора до сетевого экрана и маршрутизатора. В конечном итоге все определяется таблицей передачи, пример которой представлен на рис. 65.

Контроллер обменивается информацией с «подконтрольными» ему коммутаторами OpenFlow с помощью одноименного протокола. В рамках стандартной модели SDN задача коммутаторов — без задержек передавать пакеты с одного порта на другой, осуществляя некоторую обработку в соответствии с правилами. В ответ на запрос коммутатор может сообщить контроллеру о своих возможностях и конфигурации, а также сигнализировать об изменениях в своем состоянии, например, о потере канала или возникновении ошибки. Но в остальном коммутатор полностью полагается на контроллер. Коммутатор не знает ни о топологии сети, ни даже о своих непосредственных соседях.

	Порт комм.	MAC src	MAC dst	Eth.type	VLAN ID	IP src	IP dst	Proto/ sport	Proto/ dport	Действие
Коммутатор	*	*	00:1F:...	*	*	*	*	*	*	Port6
Сетевой экран	*	*	*	*	*	*	*	*	22	Отбросить
Маршрутизатор	*	*	*	*	*	*	198.51.100.1	*	*	Port6
Коммутация потоков	Port3	00:20:...	00:1F:...	0800	Vlan1	192.0.2.10	198.51.100.1	6/27089	6/80	Port6

Рис. 65. Примеры реализации различных специализированных функций коммутатора с помощью правил таблицы передачи (* обозначает любое допустимое значение соответствующего поля).

[47] см. https://www.opennetworking.org/sdn-resources/onf-specifications

Задача определения «общей картины» и конфигурирование коммутаторов возлагается на контроллер, а точнее, на приложения, его использующие. Это они задают топологию сети и рассчитывают оптимальные маршруты. С помощью контроллера они устанавливают нужные правила, следят за состоянием устройств, осуществляют мониторинг трафика и сбор статистики.

Другими словами, OpenFlow дает базовые функции управления любым аппаратным обеспечением — коммутаторами. А вот с программной начинкой SDN ситуация более фрагментарна. Хотя многие ведущие производители сетевого оборудования, и в первую очередь — производители коммутаторов, заявляют о своей поддержке OpenFlow, программный интерфейс от контроллера к приложениям либо вообще недоступен, либо является собственной разработкой производителя.

Это, безусловно, не позволяет архитектуре SDN полностью раскрыть свой инновационный потенциал, поскольку делает невозможной разработку «сетевых мозгов», независимых от производителя контроллера. На рынке коммутаторов присутствие OpenFLow наиболее заметно. Стратегия ведущих производителей, в том числе Brocade, BigSwitch, IBM, HP и NEC, включает развитие SDN на основе OpenFlow. Правда, OpenFlow пока что реально поддерживают лишь отдельные модели.

Наиболее впечатляющий пример использования OpenFlow на практике — распределенная внутренняя сеть Google, обеспечивающая обмен данными между датацентрами (так называемая сеть G-scale), показанная на рис. 66. Для построения этой сети компания была вынуждена разработать собственные коммутаторы, но обмен данными между ними и контроллерами основан на OpenFlow.

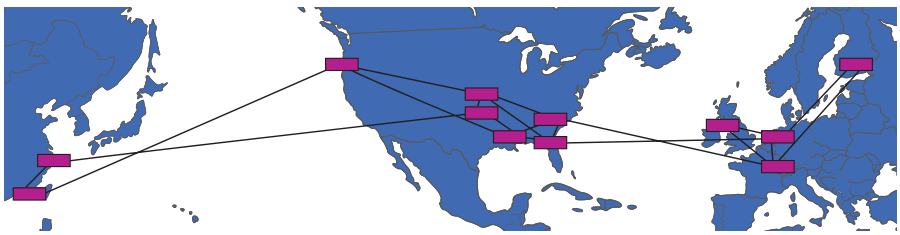

Рис. 66. Распределенная сеть Google (G-scale), использующая протокол OpenFlow.
Источник: доклад Urs Hoelzle, Google на конференции Open Networking Summit, апрель 2012 г. (https://www.segment-routing.net/images/hoelzle-tue-openflow.pdf)

Программирование системы маршрутизации — I2RS

Внедрению технологии SDN также препятствует отсутствие стандартного подхода и программного интерфейса верхних уровней. Задача становится еще сложнее, если мы говорим о миграции традиционной сети в SDN. Особенно если эта сеть обеспечивает услугу третьего уровня — маршрутизацию пакетов.

Подобно многим новым архитектурным решениям, концепция SDN (и особенно в исполнении OpenFlow) вызывает некоторое неприятие и у сетевых инженеров, и у администраторов, и у производителей сетевого оборудования, особенно маршрутизаторов.

Для первых SDN означает необходимость переключения на новый стиль управления сетью, если не переквалификацию в программиста. При этом неочевидно, что накопленный опыт разработки, управления и отладки сложных сетей может быть эффективно перенесен на новую платформу. И хотя не все задачи могут быть решены через систему управления сетью, или NMS[48], SDN представляется лишь как одна из подобных метасистем.

Производители маршрутизационного оборудования видят в SDN/OpenFlow угрозу собственной конкурентоспособности. Найдется хотя бы несколько компаний, оборудование которых переключает пакеты быстрее, чем, скажем, маршрутизаторы Juniper. Наверное, немало компаний смогут разработать программное обеспечение лучше, чем, например, Cisco. Эффективное объединение этих компонентов в одном продукте и громадная внедренная база — явное преимущество, которое может быть утрачено с приходом SDN, когда потребитель получит возможность сам выбирать и комбинировать лучшее сетевое программное обеспечение с лучшим аппаратным решением. Кстати, именно поэтому и Cisco, и Juniper предлагают SDN в собственной закрытой упаковке, означающей, по существу, улучшенный (по сравнению с пользовательским интерфейсом командной строки!) программный интерфейс для своих маршрутизаторов. Возвращаясь к вопросу миграции к новой технологии и архитектуре, отметим: лучшая программируемость сети — это уже существенный шаг вперед. Особенно если он позволяет решить некоторые насущные проблемы сетевых администраторов.

Попытки решить эту задачу предпринимались давно. Помимо повсеместно присутствующего интерфейса командной строки, сегодня существует ряд механизмов, позволяющих программно взаимодействовать с маршрутизатором.

Наиболее распространенным является протокол SNMP[49], который позволяет получать от устройства информацию о его состоянии и конфигурации, а также различную статистику. Для этого каждое устройство содержит так называемую базу управляющей информации, или MIB[50], где хранятся параметры, доступные по SNMP. Обычно SNMP используется для сбора статистики и мониторинга за состоянием устройств и их конфигурацией. Однако эта система

[48] https://ru.wikipedia.org/wiki/Система_управления_элементами_сети

[49] Simple Network Management Protocol, простой протокол управления сетью, http://ru.wikipedia.org/wiki/SNMP

[50] Management Information Base, http://ru.wikipedia.org/wiki/Management_Information_Base

крайне редко применяется для конфигурации устройства. Отсутствие необходимых функций, таких как возможность «отката», понятия тестовой конфигурации и т.п., не позволяет использовать ее в этом режиме в производственных условиях.

Другой сетевой протокол конфигурации, NetConf[51], также разработанный в IETF, снимает большинство ограничений SNMP, однако его применение ограничено ввиду отсутствия стандартной модели данных.

Система ForCES[52] предлагает принципиально другой метод конфигурирования маршрутизатора. Эта архитектура, разработанная в IETF еще в 2004 году, так же как и в случае OpenFlow, предусматривает разделение монолитного устройства на управляющий и передающий элементы[53]. Однако программный интерфейс относится к уровню передачи, тем самым не позволяя использовать «интеллектуальную начинку» маршрутизатора — логику протоколов маршрутизации и построения маршрутизационной таблицы.

Чтобы решить задачу обеспечения программного интерфейса к системе маршрутизации устройств, в IETF была создана Рабочая группа I2RS[54].

Подход I2RS направлен на использование существующей информации, собранной самим маршрутизатором о топологии сети, ее состоянии, а также о собственном состоянии и конфигурации. Ограничиваясь интерфейсами уровня системы маршрутизации, I2RS также использует существующие средства преобразования маршрутизационной информации в таблицы передачи FIB и RIB. Общая архитектура I2RS приведена на рис. 67.

Для иллюстрации возможностей I2RS рассмотрим один из примеров возможного применения — установку статического маршрута. Это часто встречающаяся задача конфигурации сети, особенно при инжиниринге трафика. Традиционные способы включают использование интерфейса командной строки и MIB, но они оба не поддерживают программного интерфейса. Обеспечивая доступ к RIB и FIB, I2RS позволяет решить эту задачу на программном уровне.

Другим примером является возможность программного доступа к функциям «policy-based routing»[55], а именно к правилам обработки определенного трафика.

---

51 http://ru.wikipedia.org/wiki/NETCONF

52 RFC 7409: Forwarding and Control Element Separation (ForCES) Packet Parallelization, URL: https://www.rfc-editor.org/rfc/rfc5155

53 Более подробно о различиях между OpenFlow и ForCES см. https://datatracker.ietf.org/doc/draft-wang-forces-compare-openflow-forces

54 Interface to the Routing System, http://datatracker.ietf.org/wg/i2rs/

55 PBR, http://en.wikipedia.org/wiki/Policy-based_routing

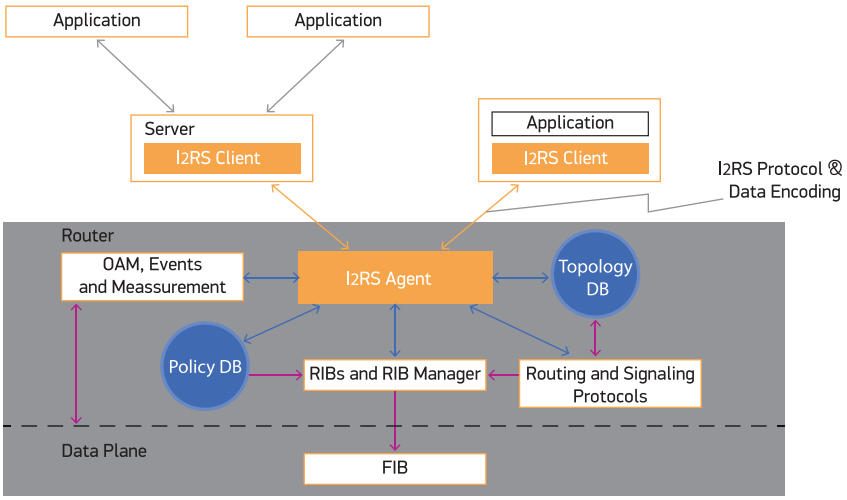

Рис. 67. Архитектура системы I2RS; желтым цветом обозначены элементы и протоколы I2RS (с разрешения A. Farrel, Old Dog Consulting).

При этом речь идет не только о доступе, например, к записям установки маршрута или изменениям правила. I2RS дает возможность получать информацию от системы маршрутизации — в частности, данные RIB или текущую конфигурацию и состояние отдельных элементов. Это не менее важно, особенно в контексте моделирования топологии сети.

Будущее SDN

На сегодняшний день SDN — это архитектурная концепция и маркетинговый термин, объединяющий множество закрытых решений построения и управления сетью. Как архитектура, SDN привлекает своей концептуальной простотой, открывая громадный потенциал для построения более сложных систем на ее основе. Кроме того, декомпозиция управляющих и передающих элементов сети открывает новые горизонты для инноваций. Стремительному развитию этой технологии может способствовать стандартизация всех интерфейсов SDN и, как следствие, создание открытой архитектуры, когда потребитель сможет самостоятельно комбинировать продукты различных производителей: коммутаторы, контроллеры и программное обеспечение управления сетью.

Сегодня стандарты — протоколы и модели данных — существуют только для «южного» интерфейса: между контроллером и коммутаторами. Стандартизация «северного» интерфейса обеспечит доступ на рынок независимым разработчикам программного обеспечения. Определение стандартного интерфейса между контроллерами позволит различным сетям взаимодействовать между собой, открывая возможности построения SDN, охватывающих несколько независимых провайдеров.

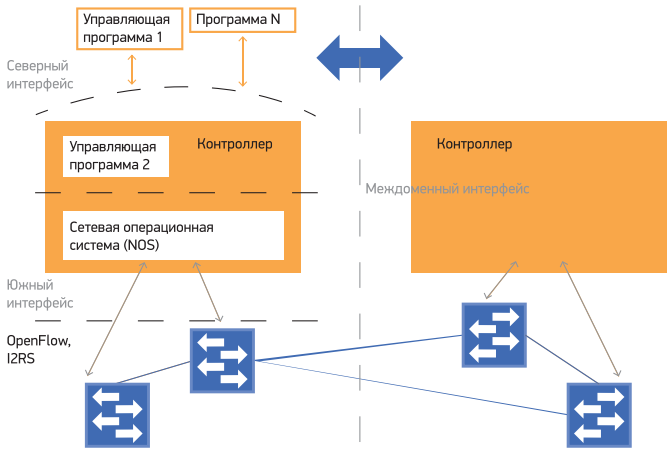

Рис. 68. Интерфейсы SDN, требующие стандартизации.

Достаточно посмотреть на число организаций, участвующих в исследованиях и разработке стандартов, связанных с SDN (рис. 69), чтобы понять — работы здесь непочатый край. Этот рисунок также свидетельство тому, насколько сильна потребность в стандартных компонентах, строительных блоках SDN, способных взаимодействовать между собой. Работы ведутся во многих направлениях, и, возможно, недалек тот день, когда SDN из архитектурной концепции и маркетингового термина превратится в стандартные протоколы и технологии.

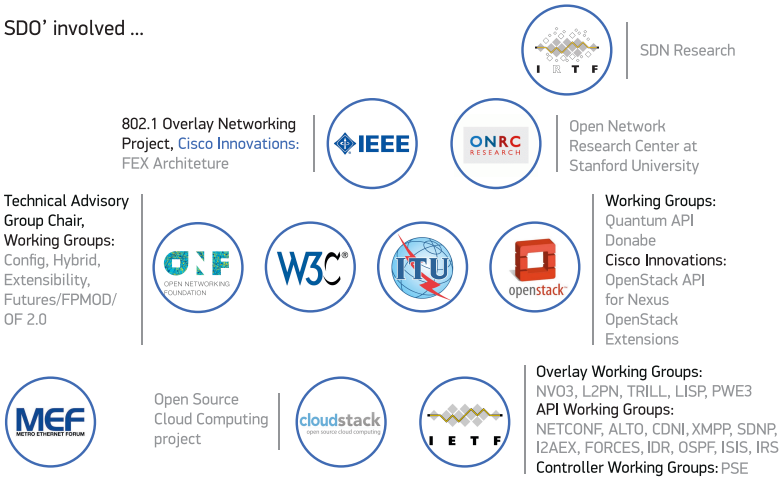

Рис. 69. Организации, участвующие в исследованиях и разработке стандартов, связанных с SDN.
Источник: доклад David Ward, Cisco, на пленарном заседании IETF84, июль 2012 г. (http://www.ietf.org/proceedings/84/slides/slides-84-iab-techplenary-3.pdf)

И кто знает — может быть, Интернет действительно станет программируемым.

Заключение

Ограниченный объем данных, называемый пакетом, в порядке эстафеты передается от отправителя к следующему узлу-маршрутизатору, затем к следующему — пока наконец не достигнет получателя. Каждый из узлов принимает решение о дальнейшем пути пакета самостоятельно, в зависимости от текущей топологии сети. Это значит, что потоки данных могут быть оперативно перенаправлены, скажем, при выходе из строя одного из узлов или «падении» канала. Передачу данных не нужно синхронизировать между всеми сетями, через которые проходит поток. Такие сети пакетной коммутации могут поддерживать любые технологии передачи данных и любые приложения. Эта новая парадигма в конце 1960-х гг. определила и продолжает определять инфраструктуру передачи данных Интернета. Управляет этой структурой протокол маршрутизации BGP, который позволяет узлам, а точнее, сетям, или автономным системам, обмениваться их представлением о топологии Интернета с другими сетями. Эта информация все время меняется, в среднем в день в Интернете происходит около 50 000 изменений топологии: подключаются новые подсети, выходят из строя каналы и маршрутизаторы, меняется топология сетей. Но эти изменения распространяются достаточно быстро, и в целом система является стабильной. Со времени внедрения протокола BGP прошло уже тридцать лет, и размер Интернета в терминах количества маршрутов вырос на три порядка. Но, как это ни странно, основные протоколы и архитектура маршрутизации в Интернете почти не изменились. Узлы по-прежнему коммутируют пакеты IP, обмен и определение маршрутов происходит с помощью протокола BGP. И удивительно, что, несмотря на столь стремительный рост, устойчивость и производительность системы по-прежнему соответствуют потребностям ее пользователей.

Однако мы уже отчетливо видим и контуры новой архитектуры: это облачные услуги, централизованные, глобально распределенные данные, доступ к которым не отличается от доступа к локальному диску компьютера. Это контент, видео, демонстрирующее высокое качество, неожиданное для сети с отсутствием поддержки качества предоставления услуг. Такая инфраструктура требует новых решений, по крайней мере во внутренней архитектуре сетей, составляющих сегодняшний Интернет.

Глобальные изменения в Интернете нелегки — будь то система адресации, имен или маршрутизации. И все-таки виртуализация Сети открывает новые возможности, в корне меняющие наши представления о топологии, производительности и местоположении ресурсов.


ПРОЦЕСС
СТАНДАРТ
ВИРТУАЛЬНЫЙ
RFC
ОБЛАКА
ВИРТУАЛЬНЫЙ
SMTP
ИНТЕРНЕТ-ТЕЛЕФОНИЯ
ОТКРЫТОСТЬ
IEEE
ТЕХНОЛОГИЯ
ПАКЕТНАЯ КОММУТАЦИЯ
FTP
ОБЛАКА
SMTP
IANA
ПРОЦЕСС
КОММУТАЦИЯ КАНАЛОВ
VOIP
ТЕХНОЛОГИЯ
НЕЙТРАЛЬНОСТЬ
VOIP
HTTP
GOOGLE
НЕЙТРАЛЬНОСТЬ
HTTP
VOIP
ПРОТОКОЛ
IETF
IANA
HTTP
ICANN
SKYPE
ПРОЦЕСС
УСТОЙЧИВОСТЬ
КОММУТАЦИЯ КАНАЛОВ
ИНТЕРНЕТ-ТЕЛЕФОНИЯ
НЕЙТРАЛЬНОСТЬ
СМАРТФОН
ЭВОЛЮЦИЯ
SIP
МАГИСТРАЛЬНЫЕ СЕТИ
СТАНДАРТ
УСТОЙЧИВОСТЬ
ИННОВАЦИЯ
VOIP
ИНТЕРНЕТ
IANA
HTTP
НЕЙТРАЛЬНОСТЬ
ПРОЦЕСС
W3C
СТАНДАРТ
ПРОЦЕСС
SKYPE
ПАКЕТНАЯ КОММУТАЦИЯ
VOIP
SMTP
ЭКОСИСТЕМА
УПРАВЛЕНИЕ
XMPP
HTTP
МСЭ
ОБЛАКА
FTP
ПРИНЦИП ПРОЗРАЧНОСТИ
МАГИСТРАЛЬНЫЕ СЕТИ
ИНТЕРНЕТ-ТЕЛЕФОНИЯ
XMPP
XMPP
ВИРТУАЛЬНЫЙ
APPLE
СТАНДАРТ
XMPP
ОТКРЫТАЯ ПЛАТФОРМА
УСТОЙЧИВОСТЬ
IGF
SKYPE
ПРОЦЕСС
АДМИНИСТРИРОВАНИЕ
WWW
GOOGLE
3GPP
APPLE
FTP
ПРОЦЕСС
XMPP
УСТОЙЧИВОСТЬ
УПРАВЛЕНИЕ
АРХИТЕКТУРА
ОТКРЫТАЯ ПЛАТФОРМА
ОБЛАКА
WSIS
НЕЙТРАЛЬНОСТЬ
ПРОЦЕСС
HTTP
ПРОЦЕСС

Глава 4

Экосистема Интернета

> Принцип постоянного изменения — возможно, единственный принцип Интернета, который должен существовать бесконечно.
>
> *RFC 1958[1], июнь 1996 г.*

История Интернета началась с четырех соединенных суперкомпьютеров — а сегодня Интернет объединил 2/3 населения мира. По данным Statista[2] на ноябрь 2023 года 5.35 миллиарда пользователей имели доступ к Интернету. В начале 2024 года Google.com насчитывал 175 миллиардов посещений ежемесячно, ежедневно передавалось 100 миллиардов сообщений WhatsApp, просматривалось пять миллиардов видеоклипов YouTube, куда каждую минуту закачивалось 500 часов нового видеоматериала! Эти факты с большими числами можно перечислять долго, и многие из них поражают воображение[3]. Я еще не закончил эту главу, а цифры уже устарели... Как долго существующая инфраструктура и технологии смогут поддерживать подобный рост? Сегодня Сеть обеспечивает передачу данных не только между оконечными устройствами, но и между облаками — распределенными виртуальными вычислительными и информационными ресурсами. Да и сами информационные

[1] RFC 1958: Architectural Principles of the Internet, URL: https://www.rfc-editor.org/rfc/rfc1958

[2] https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/

[3] https://www.internetlivestats.com

потоки все больше являются не результатом действий пользователя, а инициируются «умными объектами», которыми завтра могут стать сегодняшние лампочка и выключатель, термометр и распределительный щиток. Являются ли эти технологии по-прежнему Интернетом или это на самом деле использование проверенных технологий в другом контексте?

Фундамент Сети, неотделимой от нашей сегодняшней жизни, доказал свою необыкновенную жизнеспособность и инновационный потенциал. Хотя эта платформа обладает удивительными генерирующими свойствами, сама она является чрезвычайно консервативной — внедрение изменений в глобальном масштабе требует колоссальных усилий. На более высоком уровне мы наблюдаем образование новой генерирующей платформы, основанной на веб-технологиях и протоколе HTTP. Означает ли это ограничение конкуренции в Интернете в целом и относительное уменьшение инноваций в процессе развития Сети?

Наконец, социально-политические и экономические требования вносят изменения в технократическую модель Сети на уровне отдельного сегмента, страны или региона. Проблема в том, что предлагаемые решения не наносят видимого ущерба, позволяя добиться краткосрочного и кратковременного решения проблемы. Тем не менее, многие из таких «решений» вносят необратимые и «односторонние» изменения в основные строительные блоки Интернета, вызывая, возможно, долгосрочные проблемы. Не пора ли международному сообществу задуматься не только об «управлении», но и о защите этой уникальной платформы, подобно глобальной экологии и климату?

Давайте взглянем на экосистему Интернета более внимательно.

Открытая архитектура Интернета как основа независимой эволюции

В рамках 81-й конференции IETF в июле 2011 года была организована дискуссия «Эволюция Интернета: где находится «там» и как отсюда туда попасть?»[4]. Участники дискуссии — Бернард Абоба (Bernard Aboba, Microsoft), Марк Хэндли (Mark Handley, University College London) и Джеф Хьюстон (Geoff Huston, APNIC) — предложили свое видение траектории развития Сети. Хотя во многом их точки зрения различались, они были единогласны в одном: архитектурная чистота Сети, которая, по мнению многих, явилась основным фактором информационной революции Интернета, отчасти утрачена — и, возможно, безвозвратно. Тем не менее, с точки зрения пользователя, бум Интернета только набирает обороты: каждый день открываются новые возможности использования Сети, нас не перестают удивлять новые приложения.

[4] https://www.ietf.org/proceedings/81/index.html

Что же это — начало заката, еще не видимого простому наблюдателю, или новый виток развития этой экосистемы, основанный на принципах, отличных от родоначальных?

Не претендуя на знание ответа на этот вопрос, давайте попробуем взглянуть на эволюцию Интернета более подробно.

Генерирующие платформы

Информационная революция, свидетелями которой мы являемся, вряд ли была бы возможна без двух основных ингредиентов. Они появились в нашем мире примерно одновременно. Один из них — Интернет, другой — персональный компьютер, или ПК.

Несмотря на различия, Интернет и ПК объединяет одно важное качество: обе эти технологии являются генерирующими платформами.

Если мы взглянем на традиционную телефонную сеть, то увидим: набор услуг, предлагаемый абонентам, за более чем вековую историю изменился на удивление незначительно. Безусловно, архитектура, технологии и производительность изменились колоссально, но все это происходило «за кулисами», для поддержки основной услуги — голосовой связи между двумя абонентами. Голосовая почта, ожидание, высвечивание номера абонента, телеконференция — вот, пожалуй, и все видимые усовершенствования, ставшие доступными пользователям. Все усовершенствования разрабатывались и внедрялись «сверху вниз» телефонной компанией, которая до недавнего времени в большинстве стран являлась государственной монополией. Инновации вне ее контроля были невозможны, и всякие попытки пресекались.

Интернет и ПК оказались полной противоположностью такой среды. Вот что пишет в своей книге «Будущее Интернета и как его остановить»[5] Джонатан Зиттрейн (Jonathan Zittrain): «ПК стал революционным продуктом, потому что он притягивал инновации со стороны. То же самое можно сказать и об Интернете. Обе системы являлись генерирующими: они принимали любое нововведение, которое следовало базовому набору правил (либо было разработано для определенной операционной системы, либо работало на базе протоколов Интернета). <...> Другими словами, на компьютере могли выполняться программы, которых еще не существовало на момент его покупки. Производители ПК продавали потенциальную функциональность». Так же и в случае с Интернетом: задачей создания Сети являлось не предоставление определенного набора информационных услуг, а обеспечение глобальной связности. Выбор конкретных приложений входил в задачу самих пользователей, сеть лишь предоставляла услугу передачи данных между узлами.

[5] Jonathan Zittrain, The Future of the Internet — And How to Stop It, http://futureoftheinternet.org

Другими словами, так же, как ПК явился компьютером общего назначения с колоссальным инновационным потенциалом (в отличие от специализированных устройств), так и Интернет стал сетью общего назначения — и позволил пользователям самим определять, как и для чего она им нужна.

Передача данных — фундаментальная услуга Сети, которая сама по себе не является чем-то сверхъестественным. Но в ней нашел отражение один из основополагающих архитектурных принципов Интернета — принцип прозрачности, или end-to-end principle. Суть его в том, что большая часть функциональности реализована в оконечных устройствах — подключенных к сети компьютерах, — а не в самой сети. Сеть должна предоставлять только универсальные и нейтральные услуги, а именно — маршрутизацию и передачу пакетов данных в режиме «best effort». Сеть не дает никаких гарантий относительно параметров качества передачи и даже не гарантирует, что данные дойдут до адресата — поддержка «надежности» передачи остается за оконечными устройствами и приложениями.

Принцип прозрачности существенно упростил сетевую архитектуру, а также не позволил Сети обрести специализацию, что усложнило бы ее применение для других, пока неизвестных задач. Благодаря этому принципу, соответствующей ему архитектуре Интернета, а также открытости стандартов, на которых построена вся система, входной порог участия был (и пока остается) чрезвычайно низким, а пользователи являются «вкладчиками», привнося в Сеть свои инновационные разработки.

На самом деле это единственный архитектурный подход, который может работать в Интернете. Ведь сама Сеть образована из нескоординированной связности разнородных сетей, каждая — со своими параметрами производительности, технологиями и решаемыми задачами. Взаимоотношения между сетями в большинстве сводятся к одному из двух — отношения пиров или предоставление транзита. В редких случаях эти договоренности включают нечто большее, чем «маршрутизацию и передачу пакетов данных в режиме best effort». Универсально обязательным является только использование протоколов Интернета.

События, повлиявшие на эволюцию Интернета

Инновация инновации рознь. В своей книге «Дилемма инноватора» Клейтон Кристенсен (Clayton M. Christensen. The Innovator's Dilemma) определил два типа инноваций: поддерживающую и разрушительную.

К первой группе относятся технологические нововведения, которые хорошо вписываются в существующую траекторию развития индустрии. Такие инновации позволяют ведущим компаниям повышать эффективность своей деятельности. Ко второй группе относятся нововведения, многие из которых не имеют смысла и не соответствуют сегодняшним потребностям людей. Как правило, они остаются полуреализованными идеями — или сразу оказываются

в мусорном ведре. Но их малая толика все-таки подхватывается рынком, позволяя нам делать вещи не лучше или эффективнее, а по-другому.

Благодаря своим принципам Интернет — плодородная почва для разрушительных, или революционных, инноваций. Низкий барьер входа, независимое созидание без необходимости получения разрешений, простота распределения услуги или продукта и непосредственный доступ к сотням миллионов пользователей — все эти факторы явились решающими в удивительно стремительном развитии Интернета.

На протяжении истории Интернета можно отметить несколько вех, в значительной степени определивших его дальнейшее развитие.

Электронная почта

История глобальной электронной почты, какой мы ее знаем сегодня, начинается с ранних дней ARPANET. Вообще-то электронная почта появилась еще до возникновения Интернета и служила для обмена сообщениями между локальными пользователями компьютера (ПК тогда еще не существовало). Услуга обмена документами и сообщениями между пользователями Сети оказалась настолько востребованной, что электронная почта явилась одним из ключевых приложений, стимулировавших развитие раннего Интернета. Эта услуга была впервые документирована в спецификации RFC 561[6] в 1973 году, более чем за 10 лет до создания IETF. В 1980-е гг. электронная почта являлась основной глобальной услугой Сети, обеспечивая передачу ASCII-сообщений.

Несмотря на многообразие сегодняшних возможностей обмена информацией, электронная почта по-прежнему является одним из основных приложений в арсенале пользователя Интернета.

Коммутация пакетов

Ранние разработки технологии коммутации пакетов появились в начале 1960-х независимо в нескольких исследовательских центрах США и Великобритании. Основным ее преимуществом перед традиционной технологией коммутации каналов являлась простота и более эффективное использование канальной пропускной способности. В сетях коммутации каналов для передачи данных необходимо сначала установить соединение, и в создании и управлении соединением участвуют все узлы сети на пути предполагаемого потока данных. От сети коммутации пакетов не требуется запоминать состояние различных потоков данных, проходящих через нее. Все, что требуется от узлов в соответствии с их знанием топологии сети, — это принимать и передавать другим узлам пакеты: фрагменты данных определенного объема, заключенные в своего рода конверты с адресами отправителя и получателя.

---

[6] RFC 561: Standardizing Network Mail Headers,
URL: https://www.rfc-editor.org/rfc/rfc561

Эта технология была выбрана в качестве технологии передачи данных в сети ARPANET, где она получила дальнейшее усовершенствование. Помимо технических преимуществ, данная технология потребовала построения сети с узлами коммутации, отдельной от существовавших телефонных сетей. Все, что требовалось Интернету от телефонных компаний, — предоставление каналов связи. Более того, в сетях коммутации каналов оплата обычно производилась повременно, даже если никаких данных не передавалось, что тоже не соответствовало характеру работы Интернета.

Как следствие, Интернет развивался независимо от телефонных сетей и телефонных компаний. Впрочем, с ними в прошлом веке строители Интернета временами вели острую борьбу за независимость.

Сеть сетей

По мере роста ARPANET потребовался переход от монолитной сети к менее централизованной топологии, включавшей опорную сеть NSFNET и подключенные к ней региональные сети. Дальнейшее развитие привело к созданию нескольких опорных сетей — региональных точек обмена трафиком. Региональные сети также могли взаимодействовать между собой как пиры. Разработанный для поддержки новой архитектуры протокол маршрутизации BGP (Border Gateway Protocol) явился основополагающим для дальнейшего развития Интернета как сети сетей. Текущая версия протокола — 4, она была внедрена в 1994 году, и это последнее существенное изменение в системе маршрутизации Интернета.

Гипертекст и WWW

Электронная почта и telnet (виртуальный терминал) стали первыми приложениями, определившими популярность Интернета. Однако по мере роста информационных ресурсов Интернета обострилась необходимость организации информации в Сети. Разрабатывалось все больше приложений, таких как Archie, WAIS, Gopher, которые позволяли искать и каталогизировать информацию. Но всех победил WWW (World Wide Web), основанный на языке создания документов HTML и протоколе HTTP.

Всемирная паутина стала успешной во многом потому, что WWW не требовал определенной структуры взаимосвязи между ресурсами, как, например, Gopher, и поэтому более соответствовал самоорганизационному характеру Интернета. Другим критическим фактором успеха явилось появление в 1993 году браузера Mosaic с графическим пользовательским интерфейсом. Он придал стиль и цвет документам и, по существу, открыл мир мультимедиа для пользователей Сети.

Либерализация рынка телекоммуникаций

Конец 1990-х был отмечен существенными изменениями в телекоммуникационном рынке ряда ведущих стран, в первую очередь США и Европы. Эти перемены положили конец монопольной позиции национальных телекомов.

В 1996 году в США был принят Телекоммуникационный Акт, целью которого было создание условий для доступа новых компаний на рынок и справедливой конкуренции между ними. В Европейском союзе либерализация телекоммуникационных услуг была в основном завершена к началу 1998 года. Изменилась и задача государственного регулирования — теперь она была направлена в сторону защиты прав пользователя и предотвращения нечестной конкуренции.

Результатом явилось существенное снижение цен, особенно на международные каналы, а также появление новых игроков и услуг, в частности, на рынке канальной емкости.

Мыльный пузырь Дот-ком

Период между 1995 и 2000 гг. ознаменовался необыкновенным расцветом бизнеса и ростом числа компаний в секторе информационных технологий. В соответствии с самой популярной теорией того времени, получившей название GBF (Get Big Fast — «вырасти как можно скорее»), развитие и выживание компании зависело от максимально быстрого расширения базы ее пользователей, даже если бизнес приносил значительные убытки. Например, Google и Amazon, а также многие тысячи других компаний работали в убыток в течение нескольких лет после появления на рынке. На пике бума, чтобы успешно войти на рынок и создать значительный капитал на бирже, требовался не столько бизнес-план или предложение востребованных услуг, сколько суффикс .com в названии компании (отсюда и название «Дот-ком»). Несмотря на отсутствие прибыли, компании росли в цене, поскольку росли цены на акции. Биржевая пирамида рухнула в начале 2000 года, оставив тысячи потенциальных миллионеров с пустыми руками. Вместе с мыльным пузырем пропали и многие «стартапы».

Однако этому периоду сопутствовал небывалый уровень практически неограниченных инноваций, стимулировавший реальные прорывы в области информационных технологий. Многие сегодняшние гиганты, определяющие лицо индустрии, родились на волне .com. Изменилась и бизнес-модель многих компаний, предоставляющих информационные услуги. Предпочтительным стало бесплатное предоставление самих услуг, так любимое пользователями Интернета, и получение прибыли от рекламы и вторичных сервисов, например, технической поддержки.

Поисковые машины

Попытки организовать информацию в Интернете делались еще до появления веба. Уже упоминавшиеся Archie, WAIS и Gopher являлись навигационными маяками для блуждающих в лабиринтах Сети.

По мере роста WWW росло и число поисковых машин, отслеживающих каждое изменение в веб-пространстве. В первые годы индексировались только заголовки веб-страниц, но в 1994 году наконец-то появилась первая

полнотекстовая машина — WebCrawler. За ней последовали другие - Lycos, Magellan, Excite, Infoseek, Inktomi, Northern Light, AltaVista и Yahoo!. Начиная с 1998 года стремительную популярность обретает Google, во многом благодаря своим алгоритмам индексирования контента, позволяющим получать наиболее точные ответы.

Поисковые машины, многие из которых выросли на плодородной почве .com, открыли новые горизонты использования Интернета, превратив хаос экспоненциально растущих информационных ресурсов в уникальную базу знаний человечества.

Мобильный Интернет

Первым мобильным телефоном с доступом в Интернет был Nokia 9000 Communicator, появившийся на рынке в 1996 году. Но немедленного революционного скачка в использовании Интернета не случилось. Виной тому стали относительно низкие скорости мобильной связи, высокие цены и отсутствие контента и услуг, специально спрофилированных для использования на мобильных устройствах малого размера.

Зато революция началась в 2007 году, когда Стив Джобс (Steve Jobs) объявил о начале продаж фирмой Apple мобильного смартфона — iPhone. Обязательный безлимитный тариф на передачу данных, онлайн-магазин приложений App Store, уже хорошо знакомый пользователям компьютеров Apple и с самого начала предлагающий широкий выбор приложений, — все это определило новый стандарт работы в Интернете.

iPhone явился первым элегантным и интуитивным интерфейсом Интернета, сделав его доступным небывало широкому кругу пользователей. Но что еще более важно, каждый владелец iPhone получил свой Интернет, доступный в любое время и в любом месте. За Apple последовали и другие.

Сегодня, чтобы узнать прогноз погоды, мы нажимаем на «солнечную» пиктограмму, а не набираем адрес сайта метеослужбы. iPhone и его конкуренты изменили характер нашего взаимодействия со многими мобильными приложениями Интернета — благодаря сервисам геолокации, возможности свободно обращаться к ресурсам Сети и простоте использования.

Динамика инноваций

Прежде чем продолжить разговор о динамике инноваций в Интернете, давайте кратко остановимся на архитектуре протоколов. Традиционно она представлена четырехуровневой моделью (рис. 70).

Эта модель Интернета основана на протоколах TCP/IP и состоит из четырех уровней: канальный, сетевой, транспортный и уровень приложений. Кратко приведем их основные характеристики.

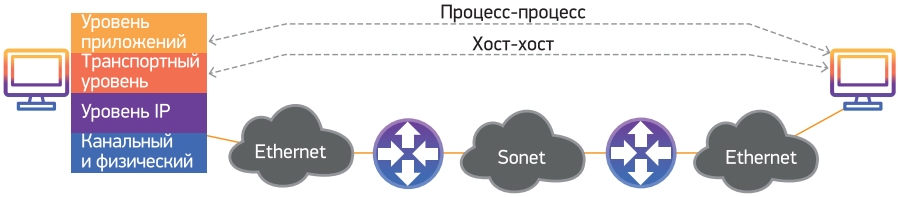

Рис. 70. Сетевая модель Интернета и принцип прозрачности на уровне транспорта (хост-хост) и приложений (процесс-процесс).

- **Канальный уровень** включает технологии и протоколы передачи данных в физической и локальной сети. Этому уровню принадлежат такие технологии, как Ethernet, Frame Relay, ATM, MPLS.
  В модели TCP/IP в этот уровень также включены стандарты кодирования и передачи сигналов в физической сети — оптическое волокно, радиосигнал и т.п.
- **Сетевой уровень** определяет передачу данных между локальными сетями, обеспечивая создание интерсетей, или собственно Интернета. Этот уровень является глобальным и универсальным — именно на сетевом уровне каждое устройство, непосредственно подключенное к Интернету, взаимодействует с другими устройствами. Технология построения различных сетей, составляющих Интернет, может быть различна, так же, как и приложения и услуги, предоставляемые в этих сетях. Однако протокол IP — основной протокол сетевого уровня — является общим знаменателем, определяющим Интернет, по крайней мере сегодня.
- Услугами протоколов **транспортного уровня** пользуются приложения, расположенные на различных хостах. Эти протоколы обеспечивают сквозную связность между хостами, а также дополнительные функции, такие как мультиплексирование виртуальных каналов, гарантированную безошибочную передачу данных, контроль пропускной способности и т.п. Основными протоколами этого уровня являются TCP и UDP. Первый из них обеспечивает обмен данными между приложениями с созданием виртуального соединения, а UDP — обмен «дейтаграммами» без создания соединения.
- **Уровень приложений,** или прикладной уровень, содержит протоколы обмена данными между приложениями, или процессами. Вот лишь некоторые наиболее значимые приложения, использующие протоколы этого уровня:
  - электронная почта: SMTP, POP, IMAP;
  - передача файлов: FTP, TFTP;
  - коллаборативные веб-платформы: HTTP, WebDAV;
  - голосовая связь: SIP;
  - обмен сообщениями: XMPP;
  - инфраструктурные приложения: DNS, DHCP, TLS/SSL.

Как видно из рис. 70, принцип прозрачности осуществляется на транспортном уровне и уровне приложений, в то время как функциональность Сети

ограничена сетевым и канальным уровнем. Интернет является открытой и генерирующей платформой во многом благодаря этой уровневой модели протоколов и принципу прозрачности.

Как заметил Тим Бернерс-Ли (Tim Berners-Lee), создатель Всемирной паутины, «я запустил сервер на одном компьютере, набрал url в браузере на другом компьютере, и браузер сделал запрос в DNS — систему, которая уже существовала долгое время, — и браузер создал соединение с сервером, используя протокол TCP, который также существовал уже много-много лет, и все это заработало, потому что [Интернет] является открытой платформой — надежной, но безразличной по отношению к приложениям, которые ее используют. И я смог создать и запустить эти программы, не спрашивая разрешения у разработчиков [TCP/IP или DNS] или у какой-нибудь организации».

Конкуренция протоколов

Как уже отмечалось, протокол IP (IPv4 и IPv6) является общим знаменателем всех устройств, подключенных к Интернету.

Пользовательские компьютеры различаются набором приложений и могут использовать различные протоколы этого уровня. Например, кто-то ограничится веб-серфингом (протокол HTTP), в то время как другой пользователь активно пользуется системами мгновенного обмена сообщениями (протоколы XMPP, SIP и др.). Некоторые специализированные устройства могут задействовать только протокол UDP на транспортном уровне.

Так же велико разнообразие используемых протоколов на уровнях ниже IP. Сегменты Сети могут использовать различные технологии — беспроводную связь, Ethernet, соединения «точка-точка». Их функциональность реализуется с помощью различных физических носителей — витая пара, радио, оптическое волокно.

Если расположить протоколы Интернета на соответствующих уровнях, то диаграмма напомнит песочные часы — шейкой которых является протокол IP (рис. 71).

Обратите внимание, как существенно различается количество протоколов на разных «слоях» диаграммы. Это различие отражает и разницу в степени возможных инноваций. По мере удаления от шейки песочных часов инновационный потенциал растет. Изменения на уровне приложений наиболее впечатляющи — каждый день появляются новые приложения, и в некоторых случаях это связано с появлением новых протоколов.

На канальном и физическом уровне, в свою очередь, инновации направлены на увеличение пропускной способности и качества сегментов Сети.

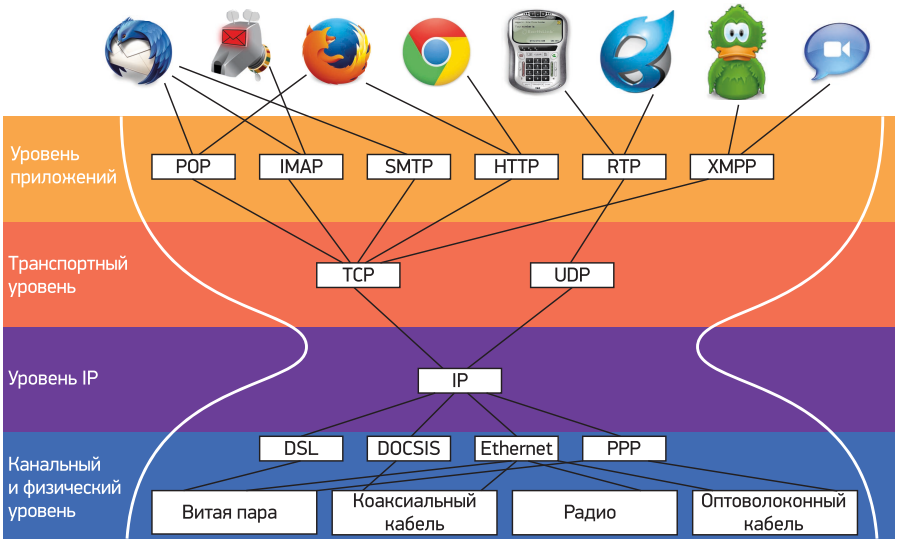

Рис. 71. «Песочные часы» стека протоколов Интернета.

Основной протокол этого уровня, IPv4, в 2020 году справил свое 40-летие! Его основной конкурент, IPv6, пока так и не занял доминирующей позиции, несмотря на многолетние усилия со стороны технического сообщества, производителей оборудования, государства и других по его внедрению.

Исследователи из Института технологии штата Джорджия (Georgia Institute of Technology) на основе численного моделирования показали, что форма стека в виде песочных часов не случайна и является результатом естественной эволюции. Используя уровневую модель, построенную на нескольких принципах взаимодействия протоколов, исследователи анализировали процесс выживаемости протокола и, как следствие, формирование определенного стека протоколов[7].

Суть модели, которую они назвали EvoArch, заключается в том, что каждый протокол зависит от услуг нескольких протоколов нижнего уровня и предоставляет услуги одному или более протоколам уровнем выше. Последнее определяет уровень «генеральности» протокола. Чем выше «генеральность» или, другими словами, чем более универсальные услуги предоставляет протокол, тем статистически больше протоколов верхнего уровня пользуются его услугами.

Чем больше протоколов используют функциональность данного протокола, тем выше его ценность. Если услуги протоколов одного уровня значительно пересекаются, эти протоколы начинают конкурировать и выигрывает протокол с большей ценностью.

7 https://phys.org/news/2011-08-internet-architecture-hourglass-future.html

Абстрактное моделирование стека различных протоколов показало, что через несколько итераций стек принимает форму песочных часов. Слабая конкуренция протоколов верхнего уровня (и, как следствие, низкая смертность) объясняется их специализацией — пересечение функциональности двух протоколов на этом уровне относительно невелико. Протоколы нижнего уровня, как правило, имеют одних и тех же потребителей услуг, но, поскольку их ценность также примерно одинакова, конкуренция и смертность и здесь невелики. Конкуренция максимальна в районе шейки часов, что приводит к выживанию значительно меньшего числа протоколов.

Этим, кстати, можно объяснить сложность внедрения нового протокола на этом уровне — неравенство в ценности настолько велико, что шансов на выживание у новичка практически нет. Реальный пример — процесс внедрения протокола IPv6, который продвигается, хотя и очень медленно, благодаря стратегической необходимости и значительным усилиям по его пропаганде, а не в результате естественной конкуренции с IPv4.

Интересно заметить, что в процессе эволюции протоколов шейка часов распространяется на протоколы более высокого уровня, например на транспортные протоколы и даже на протоколы уровня приложений. К примеру, популярность веб-технологий делает протокол HTTP универсальным и очень конкурентоспособным, а выживание других, более специализированных протоколов, проблематичным. Протокол HTTP является реальным кандидатом на новую шейку песочных часов Интернета.

Эти процессы могут иметь далеко идущие архитектурные последствия как для Интернета, так и для пользовательских устройств. Во-первых, принцип прозрачности теперь применяется на уровне HTTP, требуя от Сети лишь надежной передачи веб-трафика и не нуждаясь в более универсальных услугах. Не секрет, что сегодня протокол HTTP имеет наилучшие шансы прохождения через экраны, фильтры и другие устройства, разъедающие принцип прозрачности на уровне IP. Другими словами, веб становится новой генерирующей платформой. А браузер — новой операционной системой для интернет-терминала, в который превратится ПК.

Взгляд в будущее

Предвидение будущего Интернета — задача почти невозможная именно благодаря непредсказуемости «мутаций» этой экосистемы. Однако имеет смысл взглянуть на ряд сегодняшних тенденций, подчас противодействующих друг другу, которые оказывают влияние на будущее развитие Интернета.

Свободное развитие и регулирование

Развитие Интернета сопровождалось ростом и самоорганизацией международного технического сообщества для координирования деятельности, имеющей глобальный характер. Например, распределение адресного пространства или разработка интернет-стандартов производится организациями, деятельность которых основана

на принципах саморегулирования. Хотя функции такого рода традиционно выполнялись под контролем государства-регулятора, а для глобальных вопросов — в рамках межправительственных организаций, ни региональные интернет-регистратуры[8], ни IETF[9], ни ICANN[10] не имеют такого статуса.

Интернет стремительно развивался и благодаря этому оставался «гадким утенком» для традиционных государственных институтов. К тому же во многих странах происходили существенные преобразования по либерализации и дерегулированию рынка телекоммуникаций. Все это привело к тому, что «регулирование» по большей части обошло Интернет стороной, что, безусловно, способствовало его развитию как генерирующей платформы.

Но генерирующие технологии сами по себе не несут социального прогресса. Они стимулируют мутации в эволюционном процессе развития, изменения статус кво. Как и в любом эволюционном процессе, часть этих мутаций являются тупиковыми путями, часть же вызывают долгосрочные изменения, как хорошие, так и плохие. Интернет сегодня — это, с одной стороны, доступ к уникальной кладовой информации и невиданная до сих пор социально-информационная связность людей, а с другой — спам, вирусы, атаки на информационные ресурсы и инфраструктуру, нарушение интеллектуальных прав. Как уменьшить негативные явления, не заблокировав в то же время креативный и экономический потенциал Интернета? Ведь и хорошее, и плохое развиваются на одной и той же технологической платформе.

В то же время все острее встают вопросы использования и регулирования Интернета, которые сегодня связаны с вопросами национальной безопасности, борьбы с киберпреступностью и терроризмом, защиты интеллектуальных прав. Решение этих вопросов, несмотря на их остроту, требует обдуманного и зачастую деликатного подхода. Многие «традиционные» решения, перенесенные на интернет-почву, не только малоэффективны или просто не работают, но находятся в конфликте и разрушают основные принципы Интернета. А поскольку многие явления в Интернете не знают границ, изменения на региональном уровне имеют глобальные последствия.

Но проблемы и предлагаемые решения могут носить и чисто экономический характер. Возьмем для примера «оптимизацию» трафика в сетях доступа для повышения качества услуги или предотвращения монополизации ресурсов определенными приложениями. Как вариант, трафик реального времени получает преимущество по отношению к веб-трафику. Или, при более агрессивном сценарии, оператор навязывает пользователям свои предпочтения по отношению к информационным ресурсам, например, к собственной поисковой машине, или блокирует использование определенных приложений, например, Skype или BitTorrent.

8 https://www.nro.net
9 https://www.ietf.org
10 https://www.icann.org

Решения такого рода, помимо того, что ограничивают свободу выбора, еще и разрушают принцип прозрачности Сети, которая распространяется далеко за пределы сети оператора. Отдельные сегменты Сети начинают выполнять дополнительные функции, неожиданные для оконечных устройств, что усложняет появление новых приложений.

Эти тенденции более заметны в мобильных сетях, где «оптимизация» трафика и тарифных планов принимает более широкие масштабы. Это и введение «скоростных рядов» для поставщиков контента на основе особых договоренностей, и «zero-rate» (бесплатные) контракты, когда доступ к тем или иным ресурсам не вычитается из бюджета трафика абонента.

Чем более фундаментальные «строительные блоки» Интернета затрагивает предлагаемое решение, тем больше риск непредвиденных долгосрочных последствий, негативный эффект которых может превысить зло решаемой проблемы.

Неограниченные возможности и безопасные платформы

Сегодня нельзя не заметить, что требования пользователей к «генеративности» Интернета заметно изменились. Два десятилетия назад основными пользователями являлись сотрудники научных центров и университетов, их потребности в экспериментировании и создании новых систем и приложений были достаточно высоки. Использование Сети носило кооперативный характер и предполагало определенный уровень ответственности. Потребности сегодняшнего пользователя существенно отличаются — ему нужны готовые приложения, безопасная среда и приятный интерфейс.

Такой «стерильный» Интернет предложил пользователю Стив Джобс в 2007 году в виде iPhone. В нем все прекрасно: и внешний вид, и поистине неисчерпаемые новые возможности, открываемые нами с новыми приложениями, для установки которых, кстати, не надо проходить курс компьютерной грамотности и знать с десяток команд. Процветание iPhone непосредственно связано с инновационным потенциалом Интернета, однако iPhone изолирует нас от этой генерирующей платформы, предлагая удобное и безопасное потребление услуг, созданных на ее базе. Для многих этого достаточно. Но не надо забывать, что iPhone и вся связанная с ним инфраструктура являются лишь метаприложением Интернета, а не его безопасной альтернативой.

Фундаментальные технологии и их оссификация

Как мы обсуждали ранее, ряд протоколов являются фундаментальными для работы Интернета и в то же время обладают необыкновенно высоким порогом к изменениям. Вспомните протокол IP, образующий шейку песочных часов стека протоколов Интернета. Конкуренция на этом уровне практически невозможна, и любые изменения в существующий протокол по существу отторгаются системой.

Помимо IP, сравнимую по фундаментальности роль играют два других протокола. Это DNS, являющийся основой глобальной системы разрешения имен, и BGP,

определяющий архитектуру и функционирование глобальной системы маршрутизации Интернета. Обо всех трех протоколах мы говорили в предыдущих главах.

На уровне фундаментальных протоколов отсутствуют естественные факторы, стимулирующие инновации и изменения, которые мы наблюдаем на верхних и нижних уровнях протоколов. А значит, внедрение новых функций в глобальном масштабе — грандиозная и труднейшая задача. Говорят об оссификации, или окостенении, базовых протоколов. Это отражает фундаментальную проблему координированного внедрения в некоординируемой среде, которой является Интернет.

А такие изменения уже давно необходимы. Новая версия протокола IP — IPv6 — должна была глобально вытеснить своего предшественника задолго до исчерпания адресного пространства IPv4. Защищенность протокола DNS давно не соответствует требованиям приложений, который его используют, а расширения безопасности DNSSEC до сих пор практически не нашли применения. Наконец, система маршрутизации Интернета основана на транзитивном доверии, обеспечивая беспрепятственное распространение ошибок, большой и маленькой лжи кого-либо из более 40 000 провайдеров в глобальном масштабе. И в то же время дополнительные функции безопасности, находящиеся в настоящее время в процессе разработки в IETF, вызывают у многих сервис-провайдеров безразличие, а у некоторых — недоверие и опаску.

Такое состояние дел приводит к тому, что часть необходимых функций обеспечивается протоколами или системами на более высоких уровнях. Да, степень функциональности и безопасности подобных решений ниже предлагаемых расширений фундаментальных протоколов. Зато возможно независимое и некоординируемое внедрение с целью немедленного локального улучшения ситуации. Это делает их привлекательной альтернативой долгосрочным изменениям окостеневших протоколов. Примерами таких решений являются применение систем мультиплексирования адресов NAT, сертификатов SSL/TLS для веб-сайтов, использование эвристических фильтров маршрутов.

Разработка открытых стандартов Интернета. IEEE, IETF, W3C

Говоря о протоколах, нельзя не упомянуть о процессе разработки и стандартизации, который во многом определяет для каждого протокола перспективы его внедрения и использования. Начнем с основного вопроса:

Для чего нужны стандарты?

Ответить на этот, казалось бы, нехитрый вопрос не совсем просто. В общем смысле стандарты нужны для обеспечения совместимости и взаимодействия между элементами некоторой системы — компьютерной сети, распределенного

приложения или собственно компьютера. При этом подразумевается, что эти элементы могут быть созданы различными производителями.

В социальном и экономическом контексте стандарты способствуют свободному перемещению товаров и услуг. Уменьшая технические барьеры для создания и внедрения продуктов, стандарты помогают раскрыть новые возможности экономического развития, поощряя дифференциацию и конкуренцию между продуктами и в то же время обеспечивая совместимость и взаимодействие. Можно сказать, что стандарты являются необходимым компонентом саморегулирования промышленной отрасли.

Последнее замечание в полной мере справедливо только для открытых добровольных стандартов, принятых на основе консенсуса. Но бывают и стандарты, предписанные государством, и чисто рыночные стандарты де-факто, основанные на доминировании какого-либо игрока. Например, ранние стандарты в области информационных технологий имели именно характер «де-факто» и определялись крупнейшими производителями, такими как IBM. Говоря о стандартах Интернета, мы будем обсуждать исключительно первую категорию — ведь только открытые и добровольно применяемые стандарты имеют шанс прижиться в этой экосистеме. Здесь я имею в виду именно стандарты, а не просто решения и приложения, которые зачастую являются весьма успешными, несмотря на их, возможно, закрытый характер. Существенным является то, что стандарты являются «строительными блоками», из которых можно создать приложение — в том числе и закрытое.

Помимо положительного влияния в экономическом и социальном плане, стандартизация помогает решить и ряд более конкретных задач.

Например, в ходе стандартизации элемента или протокола решение, как правило, проходит детальную экспертную оценку и доработку. Это является сильным побудительным мотивом создания стандарта. В некоторых случаях организации по стандартизации обеспечивают полный цикл разработки решения — от начальной идеи до протокола, интерфейса или элемента.

Стандартизация решения может предоставить конкурентное преимущество изобретателю протокола или технологии. Особенно если эта технология уже воплощена в производстве. Использование процесса стандартизации в конкурентной войне — не такая уж редкость.

Все сказанное справедливо для многих отраслей, в том числе и для Интернета, о чем мы и поговорим более подробно.

Стандарты и Интернет

Когда говорят об «интернет-стандарте», в большинстве случаев имеют в виду техническую спецификацию протокола, программного интерфейса, схемы базы данных и тому подобных вещей. Как я уже упомянул, стандарт — это «строительный блок», призванный в совокупности с другими элементами создать систему или

решение. Для этого наряду со стандартами существуют и информационные документы с рекомендациями по применению стандарта или технологии для решения определенных задач. Обычно организации, занимающиеся стандартизацией, разрабатывают оба типа спецификаций. Чтобы лучше понять, каким образом и какие именно стандарты определяют функционирование Интернета, вспомним четырехуровневую архитектурную модель Сети. Каждый протокол выполняет максимально универсальные функции, необходимые для взаимодействия между устройствами на конкретном уровне. Например, Ethernet (IEEE 802.3) обеспечивает обмен данными между сетевыми интерфейсами локальной сети. Он поддерживает различные типы среды передачи (от коаксиального кабеля до оптоволокна) и скорости (от 10 Мбит/с до 100 Гбит/с). Однако хотя Ethernet и обеспечивает обнаружение ошибочных данных (фреймов), но исправление ошибок (например, путем повторной передачи) производится протоколами верхних уровней.

Такой подход обладает поистине неограниченным инновационным потенциалом, поскольку изменения протокола одного уровня не затрагивают протоколы других уровней — при условии, что интерфейсы взаимодействия между протоколами неизменны. Так, эволюция того же Ethernet происходила абсолютно независимо от протокола следующего уровня — IP. А для создания нового приложения (или протокола прикладного уровня) нет необходимости требовать каких-либо изменений от Сети.

В реальной практике, конечно, эта идеальная архитектура встречается не всегда. Иногда разработчики приложений и протоколов верхнего уровня основывают свои решения на специфических параметрах протоколов нижнего уровня, не учитывая, что протоколы могут меняться. С этим, кстати, связана одна из сложностей перехода на протокол IPv6, поскольку изменения затрагивают не только сетевой и транспортный уровни, но подчас и приложения. Зачастую нарушение межуровневого взаимодействия (англ. layering violation) связано с желанием оптимизировать производительность того или иного протокола. Дело в том, что в идеальной модели информация между уровнями должна быть минимизирована до предела, что иногда не позволяет протоколам верхнего уровня выбрать наилучший метод обработки данных. Например, если происходит потеря пакетов, то для протокола TCP это может означать перегрузку сети или плохое качество канала. TCP мог бы выбрать более адекватную стратегию — уменьшение окна передачи, уменьшение передаваемого сегмента и т.п., — но только если он получил бы дополнительную информацию с более низких уровней протоколов. Формально говоря, это является нарушением оптимальности структуры. Хотя именно на такой дополнительной информации основана система борьбы с заторами ECN и Conex, о которых мы говорили в предыдущей главе.

Другим фактором, нарушающим идеальную картину, является «неидеальность» самой Сети. Связано это в первую очередь с присутствием промежуточных устройств: трансляторов NAT[11], а также шлюзов прикладного

[11] Network Address Translator, http://ru.wikipedia.org/wiki/NAT

уровня[12]. Все эти устройства оперируют на уровнях выше IP, и таким образом, Сеть должна обладать дополнительными знаниями о протоколах более высокого уровня. В такой ситуации уже недостаточно просто изменить протокол прикладного уровня на конечных устройствах — для правильной работы эти изменения должны быть сделаны и для промежуточных шлюзов. Это существенно усложняет внедрение новых приложений и изменений.

Какими качествами должен обладать стандарт Интернета?

Стандарты Интернета играют двоякую роль. С одной стороны, они являются строительными блоками, на основе которых разработчики могут создавать приложения и распределенные системы. Например, использование стандартов TLS и SASL[13] обеспечивает требуемую защищенность приложений и услуг, а применение протокола HTTP позволяет создавать системы клиент-сервер.

С другой стороны, стандарты Интернета обеспечивают взаимодействие между компонентами, созданными различными производителями. Это открывает широкие возможности для глобального объединения независимых систем. Маршрутизаторы, пользовательские компьютеры и серверы, прочие оконечные устройства беспрепятственно обмениваются данными между собой независимо от марки производителя, сетевого провайдера или географического расположения. Единственным требованием является точное соблюдение соответствующих интернет-стандартов. Возьмем, например, систему электронной почты. Она использует протокол нижнего уровня — транспортный протокол TCP — и взаимодействует с другими системами, например, DNS, а работу этой глобальной системы обеспечивают несколько различных стандартов.

Интернет — уникальная система, основанная на кооперации между сервис-провайдерами на базе добровольного принятия открытых стандартов. Но, помимо основного требования совместимости, для успешного внедрения и использования стандарты Интернета должны обладать рядом дополнительных качеств. Давайте их перечислим.

Свободно доступные спецификации

Все соответствующие спецификации, которые необходимы для внедрения стандарта, доступны бесплатно и без каких-либо контрактных соглашений (например, соглашения о неразглашении).

Свобода от ограничений

Новые технологии могут внедряться и использоваться на основе стандарта без лицензионных сборов или ограничений.

---

[12] Application Layer Gateways, http://ru.wikipedia.org/wiki/Application-level_gateway

[13] Протокол Simple Authentication and Security Layer (SASL) стандартизован в RFC4422, https://www.rfc-editor.org/rfc/rfc4422. Он определяет способ добавления поддержки аутентификации в протоколы на основе соединения.

Открытая разработка
Все стороны, заинтересованные в новой технологии или протоколе, имеют возможность принимать участие в разработке стандарта Интернета.

Постоянное развитие
Стандарты Интернета развиваются и постоянно обновляются вместе с Интернетом с учетом новых технических требований.

Теперь, когда мы обсудили роль и необходимые качества стандартов, самое время рассказать о процессе их разработки.

Как создаются стандарты

Основной организацией по стандартизации в области Интернета является IETF. Вряд ли какая-либо другая организация или форум, занимающиеся вопросами стандартизации, могут конкурировать с IETF в области Интернета, но все же IETF не одинок в глобальном мире стандартов. Существует значительное число других организаций, занимающихся стандартами, область деятельности которых имеет отношение к Интернету. Некоторые из них в прошлом игнорировали феномен Интернета, но теперь хотят играть более существенную роль в процессе развития Сети. В их число входят W3C[14], IEEE[15], ITU[16], 3GPP[17] (http://www.3gpp.org), Unicode Consortium[18].

На рис. 72 показано, как область деятельности упомянутых организаций по стандартизации проецируется на сетевую модель Интернета.

Кратко остановимся на некоторых из них.

Internet Engineering Task Force, IETF

IETF не является организацией в полном смысле этого слова — у него нет ни штаб-квартиры, ни значительного штата постоянных работников. Почти вся работа в IETF выполняется на добровольных началах. Поэтому часто IETF называют форумом. В IETF также нет членства — любой может принять участие в работе IETF, единственным требованием является доступ к электронной почте.

IETF охватывает широкий спектр протоколов Интернета, но работа сфокусирована на протоколах уровня IP, транспортного и прикладного уровней. Соответственно, и рабочие группы разделены на так называемые предметные области: область Интернета (INT), транспорта (TSV), приложений (APP). Вопросы маршрутизации (RTG), безопасности (SEC) и эксплуатации и управления (OPS) сетей также выделены в отдельные области. Объем работ, связанных с голосовой и видеосвязью

[14] World Wide Web Consortium, https://www.w3c.org
[15] Institute of Electricaland Electronics Engineers, https://standards.ieee.org
[16] Международный союз электросвязи, https://www.itu.int/ru/
[17] 3rd Generation Partnership Project, https://www.3gpp.org
[18] https://home.unicode.org

в IP-сетях, SIP и IP-телефонии, а также с системами мгновенного обмена сообщениями, является настолько значительным, что соответствующие рабочие группы выделены в особую предметную область — в область приложений и инфраструктуры реального времени (RAI).

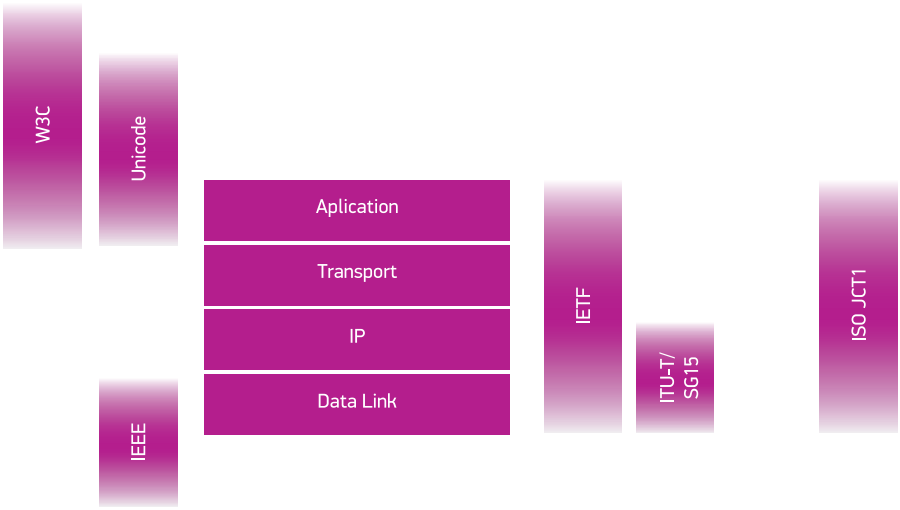

Рис. 72. Область деятельности различных организаций, форумов и консорциумов по разработке стандартов Интернета.

Структура IETF приведена на рис. 73.

Деятельность IETF осуществляется в рамках рабочих групп. По существу, рабочая группа — это список рассылки, на который может подписаться любой желающий, чтобы принять участие в процессе. Каждая рабочая группа имеет документ — «хартию», определяющую предмет работы группы и ее цели, а также рабочий план. Работу группы координируют один или два председателя.

Основная часть работы проводится в обсуждениях документов в списках рассылок, но большинство групп проводят заседания в рамках совещаний IETF, которые организуются три раза в год. Собственно, совещание IETF и состоит из заседаний рабочих групп, которые играют очень важную роль. Помимо того, что участники обсуждений общаются вживую (что весьма полезно для дальнейшей работы в виртуальном мире), заседания обеспечивают эффективное обсуждение ключевых моментов, тактику дальнейшей работы, а также позволяют подвести итоги достигнутого.

Принятие решений в рабочей группе происходит в соответствии с одним из основных принципов IETF: «грубый консенсус и работающий программный код» (rough consensus and running code). Здесь уместно рассказать о процедуре

Рис. 73. Структура IETF.

определения консенсуса на очных заседаниях рабочих групп. Традиционное голосование на таких заседаниях не принято. Председатель просит членов рабочей группы продемонстрировать свое отношение к конкретному вопросу (например, одобрить ли обсуждаемый проект) непродолжительным гудком голосом (hum). Например, сначала участников спрашивают, кто «за» предложение, а затем — кто «против». По разнице в громкости совокупного гудка председатель и определяет итоговое решение по обсуждаемому вопросу.

За общее техническое руководство деятельностью IETF и процессом разработки стандартов Интернета отвечает управляющий комитет — IESG (Internet Engineering Steering Group). Логично, что состав IESG формируется из директоров областей, которые, в свою очередь, избираются номинационным комитетом сроком на два года. Административно-организационную поддержку IETF осуществляет административное подразделение (IASA) в рамках Internet Society (ISOC), которое, впрочем, подотчетно сообществу IETF. IASA заключает контракты на оказание услуг секретариата, а также производство и издание спецификаций.

Результаты своей работы IETF публикует в виде так называемых RFC (Request for Comment или «Запрос комментариев»). Я упомянул некоторые RFC в предыдущих главах, обсуждая протоколы Интернета. Не все RFC являются стандартами: часть из них носит экспериментальный, информационный или рекомендательный характер, что отражено в статусе документа RFC. Вопросы саморегулирования (в том числе и описание самого процесса разработки стандартов) также документируются в RFC.

Что же касается тех RFC, которые являются стандартами IETF, то они проходят «многоступенчатый» процесс обсуждений и достижения консенсуса. Работа над стандартом является коллективной и обычно проводится в рамках рабочей группы. Этот процесс схематично приведен на рис. 74. Важную роль здесь играет так называемый последний звонок (Last Call, LC) — период, в течение которого участники имеют возможность высказаться в поддержку документа или отметить его недостатки. Обычно этих звонков два: один на уровне рабочей группы, а второй — общий «последний звонок», в котором участвует весь IETF.

В IETF существует также понятие «зрелости» стандарта, когда по прошествии определенного времени стандарт повышается в статусе. Как правило, это происходит в результате его обновления на основе накопленного опыта использования. В настоящее время в IETF существует два уровня: Proposed Standard и Internet Standard.

За более чем 30 лет своего существования IETF опубликовал более 7000 RFC, являющихся спецификациями приложений и протоколов, на которых построен Интернет. В работе над стандартами приняло участие более 700 различных компаний и почти 2400 авторов из более 50 стран. Хотя посещение совещаний

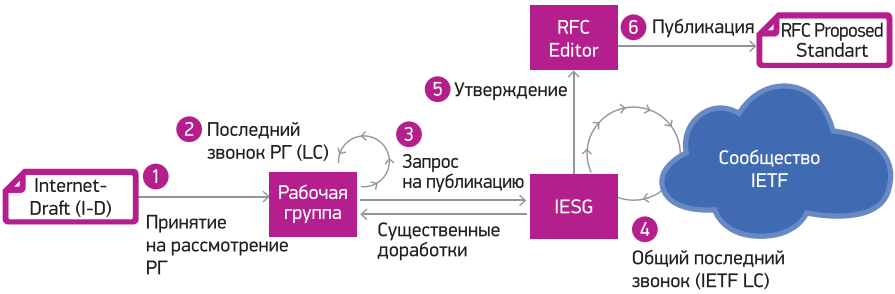

Рис. 74. Процесс стандартизации IETF.

IETF не является обязательным, три раза в год IETF собирает около 1500 участников со всего мира. Все RFC и рабочие документы (называемые Internet Draft, или ID) свободно доступны на сайте документов IETF[19].

Международный союз электросвязи, МСЭ (International Telecommunication Union, ITU)

МСЭ является учреждением ООН и работает в соответствии с конвенцией и уставом, ратифицированными государствами-членами союза. В настоящее время МСЭ насчитывает в своем составе 193 страны. Наряду с государствами в работе МСЭ принимают участие свыше 700 организаций частного сектора и академических учреждений, хотя в принятии многих решений они имеют лишь право совещательного голоса.

МСЭ охватывает широкий круг вопросов, включая распределение радиочастотного спектра и спутниковых орбит, разработку технических стандартов, а также улучшение доступа к информационно-коммуникационным технологиям в развивающихся регионах. В контексте стандартизации интерес представляет Сектор стандартизации электросвязи (МСЭ-Т).

Работа в МСЭ-Т организована в рамках так называемых исследовательских комиссий, ИК (Study Group, SG). Каждая ИК, в состав которой может входить несколько рабочих групп (РГ), координирует работу по ряду исследовательских вопросов (Questions), касающихся соответствующей темы. Например, рабочая группа по кодированию носителей в 16-й Исследовательской комиссии занимается всеми исследовательскими вопросами, относящимися к кодированию речи, аудио- и видеопотоков.

Результаты работы публикуются в виде рекомендаций. Обычно работа над рекомендациями проводится не в списках рассылки, как в IETF, а в рамках очного собрания РГ или ИК. При необходимости эксперты могут встречаться независимо от РГ или ИК в неформальной обстановке. Как правило, рекомендации доступны на сайте МСЭ, хотя рабочие документы доступны только членам союза.

[19] https://datatracker.ietf.org/doc

Процесс утверждения рекомендации также существенно отличается от процесса стандартизации в IETF. Во-первых, в МСЭ-Т в процедуре участвуют государства-члены, имеющие право окончательного голоса. Так называемая традиционная процедура утверждения, ТПУ (Traditional Approval Process, TAP), имеет очень формализованную структуру и продолжительный консультационный период (минимум три месяца), а утверждение стандарта проходит в рамках пленарных совещаний ИК, проводящихся обычно два раза в год. Возражения хотя бы одного государства-члена достаточно для того, чтобы стандарт не был утвержден. Все это удлиняет и усложняет процесс принятия решений и делает его недостаточно гибким для быстро меняющегося ландшафта информационно-коммуникационных технологий.

Однако в 2001 году процесс создания стандартов был упрощен благодаря внедрению альтернативной процедуры утверждения (АПУ). Подавляющее большинство стандартов теперь утверждается именно таким способом, а традиционную процедуру утверждения (ТПУ) проходят только те стандарты, которые имеют регуляторные последствия. Важным фактором, способствующим использованию АПУ, является возможность доработки документов в онлайн-режиме. В подавляющем большинстве случаев после начала процедуры утверждения оставшаяся часть процесса может быть завершена электронным способом без необходимости очных собраний. Схема процесса АПУ приведена на рис. 75.

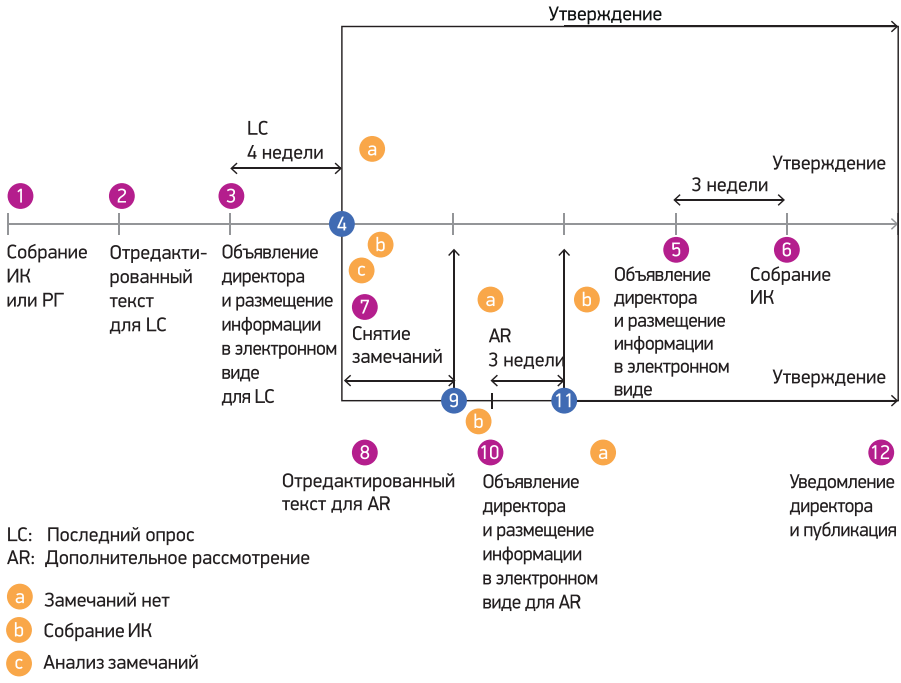

Рис. 75. Альтернативная процедура утверждения Рекомендации в МСЭ-Т.
Источник: Рекомендация МСЭ-Т A.8 (01/2024)

Многие стандарты МСЭ-Т имеют непосредственное отношение к Интернету. Например, кодеки для сжатия видео и аудио (H.264) и технологии канального уровня (DSL, WDM, ASON). Также МСЭ-Т работает над мета-вопросами — такими как Сети последующего поколения (Next Generation Networks, NGN), Интернет вещей (Internet of Things, IoT), информационная безопасность.

Консорциум «Всемирной паутины» (World Wide Web Consortium, W3C)

W3C — организация, разрабатывающая и внедряющая технологические стандарты для всемирного веба. Консорциум возглавляет сэр Тим Бернерс-Ли, основоположник WWW. Консорциум был основан в 1994 году при Массачусетском институте технологии (MIT) и объединил различные компании, заинтересованные в дальнейшей разработке технологий и создании стандартов для расширения качества и возможностей WWW. Бернерс-Ли не запатентовал свою идею, этому же принципу следует и консорциум. Его стандарты свободны от патентов и роялти, чем обеспечивается возможность их неограниченного использования.

Стандарты W3C называются Рекомендациями (Recommendation). В соответствии с процессом стандартизации, принятом в W3C, спецификация проходит четыре ступени зрелости: рабочий проект (Working Draft), рекомендация-кандидат (Candidate Recommendation), предлагаемая рекомендация (Proposed Recommendation) и, наконец, рекомендация oW3C.

Разработка рекомендации проходит в рамках рабочей группы, в которой участвуют сотрудники организаций-членов, штатные сотрудники и в некоторых случаях приглашенные эксперты. Процесс «созревания» стандарта проходит несколько степеней согласования, основанных на достижении консенсуса. Окончательное утверждение осуществляет лично директор W3C.

Рекомендация-кандидат является результатом консенсуса относительно рабочего проекта и считается готовой к реализации в ПО. Далее, предлагаемая рекомендация предусматривает, что вся предложенная функциональность была реализована и протестирована хотя бы двумя независимыми разработчиками. Наконец, рекомендация W3C является полноценным стандартом, предназначенным к широкому внедрению.

Работа W3C в основном ограничена веб-технологиями и включает следующие области.

Электронное правительство (eGovernment). Многие технологии, разработанные консорциумом, имеют прямое отношение к реализации концепции электронного правительства. Наиболее важными являются вопросы доступности и совместимости.

Языки представления контента. Сюда входит работа над новым поколением языка HTML — HTML5, поддерживающим работу с новейшими мультимедийными приложениями, а также дальнейшее развитие XML и XHTML.

Интернационализация. Поддержка национальных языков и алфавитов — одна из приоритетных областей работы консорциума.

Безопасность. Сегодня, когда широко применяется и поддерживается активный контент (например, ActiveX и Javascript), веб-браузер поистине является операционной системой, имеющей доступ к различными элементам компьютера пользователя — данным, периферийным устройствам и т.п. Не секрет, что даже простое просматривание веб-страниц может явиться причиной заражения компьютера вирусом или утечки личных данных.

Семантический веб. Это новая концепция и связанная с ней архитектура, целью которой является обеспечение большей доступности веба для компьютеров. Другими словами, семантический веб — это «веб данных». Такой подход открывает новые возможности использования Всемирной паутины в задачах консолидации данных, обнаружения и классификации информационных ресурсов, управления знаниями и т.д.

Институт инженеров по электротехнике и электронике (Institute of Electrical and Electronics Engineers, IEEE)

IEEE — это международная некоммерческая ассоциация специалистов в области техники, мировой лидер в области разработки стандартов. Стандарты IEEE работают в различных отраслях промышленности, включая энергетику, здравоохранение, информационные и телекоммуникационные технологии (ИКТ), транспорт и многие другие. В области ИКТ наиболее значительные стандарты, имеющие отношение к Интернету, разработаны в комитете IEEE 802 LAN/ MAN Standards Committee (LMSC).

Этот комитет включает более десятка активных рабочих групп (РГ), создающих и сопровождающих стандарты физического и канального уровней. Наиболее известными являются 802.3 (Ethernet), 802.11 (WLAN, или Wi-Fi) и 802.16 (Broadband Wireless Access, в частности, стандарты WiMAX).

В IEEE работа над стандартом начинается с создания так называемого запроса на авторизацию проекта (Project Authorization Request, PAR), по существу, это проект хартии будущей РГ. PAR должен продемонстрировать, что предлагаемый стандарт имеет рыночный потенциал, совместим с другими стандартами, решает определенную проблему, а также технически и экономически осуществим.

В случае, если PAR утвержден исполкомом IEEE 802, формируется РГ. В состав рабочей группы может войти любой желающий. Основным требованием является участие в заседаниях, которые обычно проводятся шесть раз в год (часть — пленарные, часть — промежуточные заседания), а также участие в голосовании по бюллетеням.

Следующая стадия — разработка проекта стандарта рабочей группой. Проект окончательно согласовывается в РГ на основе голосования по бюллетеням, для

перехода на следующую стадию требуется 75% голосов в поддержку и отсутствие серьезных возражений. После этого объявляется общее голосование, в котором может принять участие любое лицо, заинтересованное в стандарте. Необходимым условием является членство в IEEE SA (Организации по стандартизации), но можно также получить бюллетень за отдельную плату. Проект считается принятым в случае возврата как минимум 75% разосланных бюллетеней, в 75% из которых голоса отданы «за».

Если проект получает достаточную поддержку, он передается в ревизионную комиссию (RevCom) для проверки соответствия внутренним требованиям IEEE.

Наконец, по получении положительной рекомендации ревизионной комиссии стандарт окончательно утверждается Советом по стандартизации IEEE большинством голосов. Этот процесс схематично показан на рис. 76.

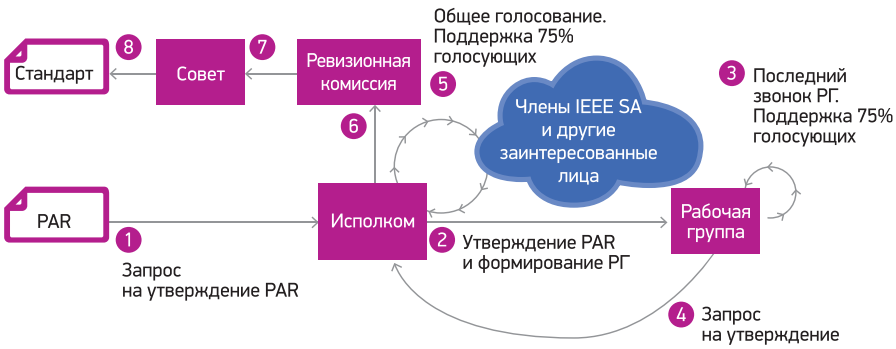

Рис. 76. Процесс разработки и утверждения стандартов в IEEE.

Все стандарты IEEE доступны онлайн или в печатном виде, но не бесплатно. В среднем стандарт стоит немногим более $100.

В заключение отметим, что исторически стандарты принимались специально созданными национальными и международными формальными организациями по стандартизации. Например, ANSI в США, DIN в Германии, Росстандарт в России. Или ISO и МСЭ, действующие в международном масштабе. Многие из этих стандартов являлись обязательными к применению, хотя сегодня в подавляющем большинстве стран это требование смягчено или полностью снято и стандарты применяются на добровольной основе.

Сегодня все большую значимость получают негосударственные и менее формальные институты стандартизации — так называемые консорциумы и форумы, особенно в области ИКТ. Два наиболее выразительных примера — W3C и IETF. Во многих случаях подобные организации не имеют официального статуса в глазах национальной организации по стандартизации, несмотря на то, что почти все

стандарты, обеспечивающие работу Интернета, были и продолжают разрабатываться именно форумами и консорциумами. Но ситуация меняется: например, Европейская комиссия работает над модернизацией политики стандартизации в области ИКТ, которая позволит официально признавать стандарты форумов и консорциумов. Многие государства также озабочены вопросом наиболее эффективной интеграции таких стандартов в национальную систему стандартизации. Ведь в противном случае придется заново изобретать велосипед.

Эволюция системы принятия решений в Интернете. ICANN, IGF

За относительно непродолжительное время своего существования Интернет прочно вписался в жизнь общества. Тем не менее, модель взаимодействия между различными его компонентами настолько отличается от традиционных представлений о координации глобальной системы, что дискуссия об управлении Интернетом продолжается уже больше десятка лет. Впрочем, без существенных результатов.

И действительно, трудно представить, что такая сложная система, охватывающая страны и континенты, включающая в себя различные культуры и технологии, может функционировать и развиваться только за счет самоорганизации. Но если все-таки необходима координация, если нужно управление, — то чем конкретно? Этот вопрос, безусловно, волнует правительства государств, как и вопрос об их роли в процессе принятия решений.

Система принятия решений и управления Интернетом в международном обиходе имеет емкий термин Internet Governance. Чтобы лучше понять ее проблематику, нам опять придется обратиться к истории.

Разумный хаос

Наверное, не будет преувеличением сказать, что до второй половины 90-х гг. прошлого века принятие решений в Интернете осуществляло техническое сообщество — инженеры, ядро которого составляли участники IAB и IETF. Ключевой фигурой являлся научный сотрудник Института информатики (ISI) Университета Южной Калифорнии (USC) Джон Постел (Jon Postel). Джон руководил, а по существу сам являлся IANA — организацией по присвоению номеров, на которой мы остановимся чуть позже.

Достаточно посмотреть на одну из первых политик присвоения доменных имен и добавления доменов верхнего уровня, описанную в RFC 1591[20] в 1994 году. Этот документ, в частности, определяет пять доменов верхнего уровня — .edu,

[20] RFC 1591: Domain Name System Structure and Delegation, URL: https://www.rfc-editor.org/rfc/rfc1591

.com, .net, .org, и .int[21] — в качестве «глобальных». Интересно заметить, что многие вопросы, обсуждение которых сегодня занимает годы консультаций и совещаний, в RFC 1591 решались парой параграфов простых правил.

Ситуация начала стремительно меняться с расширением Интернета и его коммерциализацией. В центре этой коммерциализации находилась DNS, а ключевую роль начала играть частная компания, хорошо связанная с оборонными агентствами США, — Network Solutions, Incorporated, или NSI.

В главе 2, говоря о глобальной системе имен Интернета, мы упомянули центральную регистратуру InterNIC, сопровождавшую корневую зону DNS и отвечавшую за регистрацию имен в доменах верхнего уровня .com, .org, .net и т.д. До 1993 года, когда NSF принял на себя финансирование «гражданской» части Интернета, поддержка InterNIC (тогда называвшегося DDN-NIC) осуществлялась Министерством обороны США. Функции DDN-NIC начиная с 1991 года выполнял подрядчик — компания NSI.

В 1993 году NSF объявила тендер на осуществление регистрационных услуг InterNIC, победителем которого стала все та же компания NSI. Контракт[22] 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 году NSF изменила положения договора, позволив NSI установить плату в $50 в год за доменное имя второго уровня[23].

Бум приватизации и взрывного развития Интернета набирал обороты, и регистрация приняла глобальный характер. Соответственно, росли и доходы NSI.

Вопросы относительно 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. Ну а правительства государств постепенно понимали, что эти вопросы граничат с их суверенными интересами.

---

[21] Изначальные домены верхнего уровня — .edu, .com, .net, .org и .mil — были документированы в «официальной политике IAB и DARPA» RFC 920 (Domain Requirements, URL: https://www.rfc-editor.org/rfc/rfc920).

[22] https://archive.icann.org/en/nsi/coopagmt-01jan93.htm

[23] https://archive.icann.org/en/nsi/coopagmt-amend4-13sep95.htm

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 году. В отсутствии NSF будущее управления корневой зоной было по меньшей мере непонятным.

В главе 2 было упомянуто о некоторых попытках разрешить эту ситуацию, а именно — речь шла о Проекте Постела (Draft Postel), разработанном Джоном Постелом в 1996-м, и о 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[24] 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

[24] https://web.archive.org/web/19980415071855/gtld-mou.org/draft-iahc-gTLDspec-00.html

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 года в Женеве. Более 200 организаций[25] (подписали протокол, включая Международный союз электросвязи (МСЭ), уже упомянутый 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 ситуацию был вынужден вмешаться Ира Магазинер (Ira Magaziner), советник президента Клинтона. В июле 1997 года агентство NTIA (Национальная администрация по телекоммуникациям и информации США — National Telecommunications and Information Administration) Министерства торговли США опубликовало проект приватизации DNS для публичного обсуждения[26]. 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 a.root-servers.net[27].

---

25 https://www.itu.int/newsarchive/projects/dns-meet/KeynoteAddress.html

26 https://www.ntia.gov/files/ntia/publications/022098fedreg.txt

27 Сегодня a.root-servers.net — всего лишь один из 13 вторичных авторитетных серверов, обслуживающих корневую зону. Содержимое этой зоны все серверы получают от так называемого спрятанного мастер-сервера, обслуживаемого Verisign, после приобретения NSI в 2000 г.

Значение этого проекта, как и последующего «Зеленого документа» (Green Paper)[28], не всеми было понято с достаточной ясностью. Комитет IAHC продолжал действовать согласно изначальному плану, и в октябре 1997 года 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 энтузиазма в правительстве США, и Магазинер настойчиво попросил Постела прекратить его.

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 года ICANN была зарегистрирована как некоммерческая общественная корпорация в штате Калифорния, США.

Корпорация Интернета по распределению адресов и номеров, ICANN

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. Серьезной потерей для ICANN стала внезапная смерть Джона Постела, которая «украла у организации ее интеллектуальный центр,

[28] В США и некоторых других странах термин «Зеленая книга» (Green Paper) обозначает временный консультационный отчет для публичного обсуждения, но без каких-либо обязательств к последующим действиям. Это, как правило, первый шаг на пути изменения законодательства или государственной политики. За «Зеленой книгой» может последовать «Белая книга» (White Paper) — официальный документ, разъясняющий цели и суть предполагаемого изменения или новой политики и предусматривающий в то же время опрос общественного мнения.

основной источник технического ноу-хау, значительную часть институциональной памяти и добрую долю ее правомерности»[29].

Но 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 года. 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 независимой 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 выполняли функции технического обслуживания корневой зоны — внесение изменений и предоставление зоны корневым серверам, — а также функцию регистратуры доменов .com, .net и .org. Этот договор[30] сегодня содержит 32 поправки, отражающие 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[31], 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.

Наконец, в начале 2000 года 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.

---

29 Froomkin A. Michael. «Wrong Turn in Cyberspace: Using ICANN to Route Around the APA and the Constitution», 50 Duke Law Journal. 17 (2000) (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=252523)

30 Verisign Cooperative Agreement, https://www.ntia.gov/page/verisign-cooperative-agreement

31 Registry Agreement, https://archive.icann.org/en/nsi/coopagmt-amend19-04nov99.htm

В то же время создание ICANN обозначило важную веху в системе координации централизованных функций Интернета — администрирование корневой зоны, глобального пула адресного пространства IP и уникальных параметров протоколов Интернета.

В духе стратегии приватизации Интернета была создана формально независимая от правительства США частная корпорация. В духе общей тенденции глобализации Интернета в процесс принятия решений относительно администрирования этих критических ресурсов были вовлечены глобальные сообщества. Часть этих сообществ, например, сообщество разработчиков протоколов IETF, адресное сообщество вокруг РИРов, сформировались в ходе эволюции Интернета и функционировали в соответствии с нормами и политиками, проверенными практикой. С именами дело обстояло гораздо сложнее. Монопольная позиция NSI в отношении администрирования корневой зоны и основных доменов верхнего уровня была причиной отсутствия организованного сообщества, объединенного общими интересами, согласованными правилами и политиками. Для заполнения этого пробела сообщество имен в рамках структуры ICANN было организовано следующим образом:

- Представители администрации национальных доменов были объединены в организацию поддержки национальных доменов — ccNSO.
- Для общих доменных имен также была создана организация поддержки — gNSO, однако ее структура оказалась куда более сложной и отражала широкий круг заинтересованных сторон: коммерческих организаций (бизнес, интеллектуальная собственность, интернет-сервис-провайдеры), некоммерческих организаций, а также регистратур и регистраторов.
- Наконец, интересы интернет-пользователей и правительств государств были учтены путем создания консультационных комитетов — ALAC (At-Large Advisory Committee) и GAC (Government Advisory Committee).

Описанная выше структура представляет органы, принимающие решения и определяющие политику, на основании которых IANA создает и обслуживает соответствующие реестры. При этом IANA, оставаясь подразделением ICANN, выполняет административные функции и не принимает активного участия в разработке политик.

Здесь самое время кратко остановиться на самой IANA.

Администрирование уникальных параметров Интернета. 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 понять место IANA в системе принятия решений, полезно рассмотреть различные роли, присутствующие в каждой из ее функций.

Определение политик, обслуживание регистратур и надзор

Для того чтобы лучше понять роль участников «экосистемы IANA» (рис. 77), стоит подробнее остановиться на трех основных аспектах глобальной регистратуры — политике, исполнении и надзоре.

Задачей всех вышеописанных структур принятия решений в отношении IANA является **выработка соответствующих политик,** начиная от создания специальных реестров и заканчивая правилами внесения изменений.

Так, сообщество IETF в процессе разработки протокола определяет необходимость создания реестра и правила его сопровождения. Эта «политика» документируется в разделе IANA considerations соответствующего стандарта. Документ RFC 5226 «Guidelines for Writing an IANA Considerations Section in RFCs»[32] содержит практические рекомендации по разработке таких правил и их документации в этом разделе. Замечу, что ICANN в плане разработки или утверждения таких правил не играет никакой роли.

В отношении номерных ресурсов — IP-адресов и номеров автономных систем — дело касается утверждения глобальных политик — правил управления глобальным пулом номерных ресурсов, который администрирует IANA. Сами политики разрабатываются РИР-сообществами, но затем утверждаются ASO (Address Supporting Organization, Организация поддержки адресов) ICANN. Заключительным шагом является ратификация глобальной политики советом директоров ICANN. Эти политики действительны только в отношении реестров так называемых глобальных номерных ресурсов, делегированных РИРам для последующего распределения. Часть адресного пространства (и соответствующие

[32] RFC 5226: Guidelines for Writing an IANA Considerations Section in RFCs, URL: https://www.rfc-editor.org/rfc/rfc5226

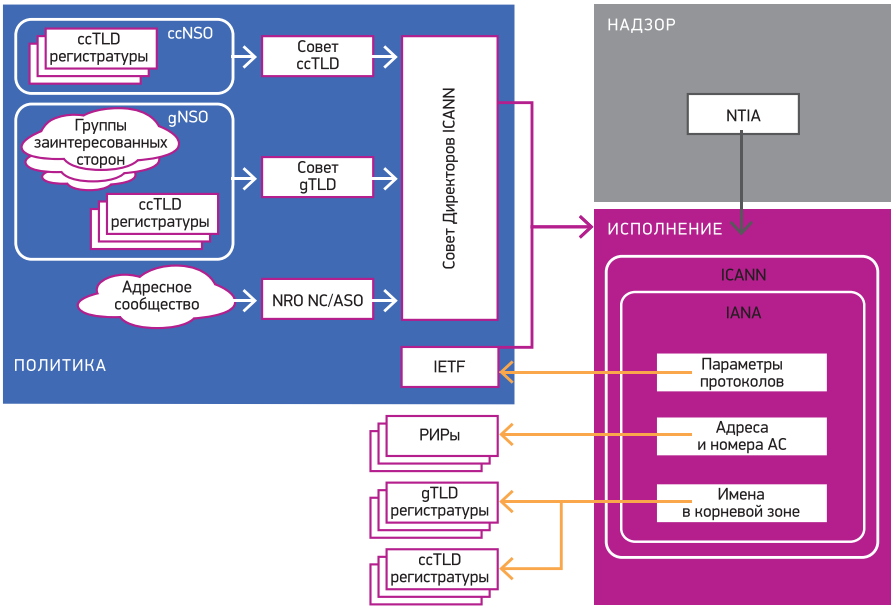

Рис. 77. Взаимодействие функций разработки политик, исполнения и надзора в «экосистеме» IANA до передачи ключевых функций.

реестры) обслуживается в соответствии с политикой IETF и является либо зарезервированной, либо предназначенной для специального использования[33].

Разработка политики в отношении доменов верхнего уровня, общих и национальных, происходит в рамках соответствующих организаций поддержки и координируется соответствующими советами — советом ccNSO и советом gNSO. Политики утверждаются советом директоров ICANN. Замечу, что большинство политик и рекомендаций, разработанных этими сообществами, непосредственно к IANA отношения не имеют. Эти политики регулируют деятельность регистраторов имен второго уровня в общих доменах верхнего уровня. Основные же правила, относящиеся к администрированию корневой зоны, — либо внешние (как, например, допустимые имена национальных доменов, определяемые стандартом ISO 3166-1), либо проходят как инструкции и рекомендации, разработанные самой ICANN (например, процесс переделегирования).

Сама же IANA выполняет сугубо **административно-исполнительную функцию,** а именно обслуживает запросы на изменение соответствующих реестров:

[33] См. например https://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml

- для доменных имен — собственно корневую зону, файл хинтов и базу данных доменов верхнего уровня, так называемую whois, содержащую, помимо прочего, контактную информацию администратора домена. В зону ответственности IANA входит регистрация имен в домене .int, а также информация и реестры, относящиеся к DNSSEC, — ключи точек доверия, материалы церемоний подписания и т.п.[34];
- для номерных ресурсов — реестры глобальных адресов и реестры номеров автономных систем[35];
- для параметров протоколов — различные реестры для протоколов, разработанных IETF[36].

До октября 2016 года функцию надзора выполняло правительство США в лице NTIA, поскольку именно это агентство министерства торговли являлось держателем договора на предоставление услуг IANA.

Для доменных имен агентство NTIA выполняло еще и дополнительную надзорную функцию, а именно утверждало любые изменения в корневой зоне и базе данных whois. Целью этой функции является проверка факта, что при обслуживании запроса на изменение ICANN следовала существующим правилам и процессу. Без этого утверждения Verisign, которая, собственно, технически обслуживает и публикует корневую зону, изменений в нее не внесет.

Эта схема существенно изменилась в результате передачи ключевых функций IANA мировому интернет-сообществу. Об этом мы поговорим в следующем разделе.

Передача ключевых функций IANA

14 марта 2014 года NTIA опубликовало заявление о намерении передать ключевые функции глобальной системы имен DNS в руки мирового сообщества[37]. Заголовок заявления обозначил главную проблему, которую агентство NTIA предполагало решить, — исключение явного асимметричного присутствия правительства США в процессе принятия решений в отношении корневой зоны глобальной DNS — надзорной функции, о которой мы только что говорили. Однако проблема оказалась масштабнее, поскольку по исторической случайности правительство США, помимо уникальных имен корневой зоны DNS, оказалось формально вовлечено в администрирование других уникальных параметров Интернета: протоколов и адресов.

В своем заявлении NTIA обозначила процесс разработки плана передачи и основные требования к нему. ICANN поручалось созвать заинтересованные

34 https://www.iana.org/domains
35 https://www.iana.org/numbers
36 https://www.iana.org/protocols
37 https://www.ntia.doc.gov/press-release/2014/ntia-announces-intent-transition-key-internet-domain-name-functions

стороны глобального интернет-сообщества для разработки плана. Результат должен быть поддержан широкими массами и удовлетворять четырем основным принципам:

1. Поддерживать и укреплять модель «мультистейколдеризма».
2. Обеспечивать безопасность, стабильность и прочность системы DNS.
3. Удовлетворять потребности и ожидания глобальных потребителей и партнеров услуг IANA.
4. Поддерживать открытость Интернета.

Отдельной строкой было выделено условие: роль NTIA не может быть передана государственной или межгосударственной организации.

В результате нескольких месяцев интенсивных дискуссий интернет-сообществом был согласован процесс создания плана передачи, согласно которому «операционные сообщества» — протоколов, номерных ресурсов и имен — должны разработать части плана, соответствующие их области деятельности, а созданная Координирующая Группа (IANA Stewardship Transition Coordination Group, ICG) должна скомпилировать общий согласованный план, при этом не добавляя ничего от себя.

В изначально установленный срок, 15 января 2015 года, были поданы два предложения — от протоколов и номерных ресурсов. Предложение от имен поступило пять месяцев спустя — 25 июня 2015 года. Наконец, к концу октября того же года группа ICG завершила работу над консолидированным предложением. Однако уже к концу лета было очевидно, что изначальный план нереалистичен, и в сентябре 2015 года NTIA продлило контракт еще на один год, установив очередную целевую дату — сентябрь 2016.

Только 10 марта 2016 года Координирующая Группа ICG смогла одобрить общий согласованный план передачи IANA и направить окончательное предложение для передачи NTIA через Правление ICANN. Днем позже Правление ICANN постановило передать это предложение на рассмотрение NTIA.

В тот же день NTIA опубликовало на своем сайте заявление руководителя NTIA Лоренса Стриклинга (Lawrence E. Strickling)[38]. В частности, он отметил: «Теперь NTIA начнет процесс рассмотрения этого предложения — мы надеемся, в течение 90 дней, — чтобы определить, соответствует ли оно критериям, которые мы определили, когда объявили о передаче».

В июне 2016 года NTIA подтвердило, что согласованный план соответствует четырем основным принципам, а двумя месяцами позже, в августе, сделало заявление о намерении не продлевать контракт[39]. 30 сентября 2016 года за-

[38] https://www.ntia.gov/blog/reviewing-iana-transition-proposal
[39] https://www.ntia.gov/blog/update-iana-transition

вершился срок действия текущего контракта на предоставление услуг IANA, 16 лет спустя после его заключения.

Согласно плану, функции IANA перешли к оператору IANA, выполняющему их по контракту с ICANN. В настоящее время таким оператором является PTI (Post-Transition IANA), некоммерческая организация, аффилированная с ICANN. Надзорная функция перешла к сообществам IETF (IESG/IAB), Региональных Интернет-Регистратур (Review Committee) и самой корпорации ICANN (Customer Standing Committee, CSC), для которых PTI выполняет административные функции в отношении протоколов, адресов и имен соответственно. Новая структура управления показана на рис. 78.

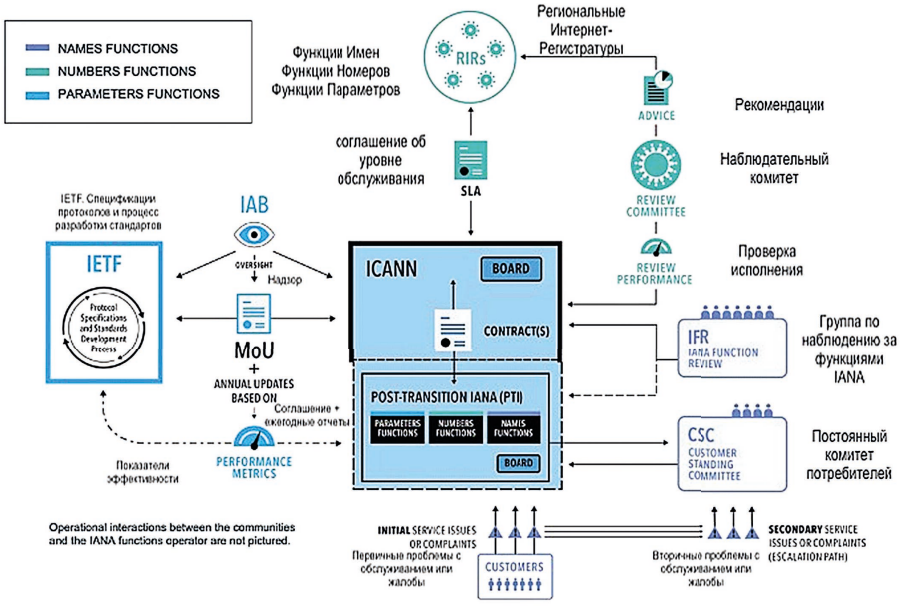

Рисунок 78. Структура IANA после передачи ключевых функций.

Всемирный саммит по вопросам информационного общества, WSIS

Вернемся, однако, обратно к 1998 году, к моменту создания ICANN. Как мы уже видели, хотя ICANN являлась формально независимой частной организацией, правительство США сформировало сеть договоров и протоколов о взаимопонимании, цементируя 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 году[40], 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.

Всемирный саммит по вопросам информационного общества (World Summit on the Information Society, WSIS) включил в себя две конференции, организованные под эгидой ООН, — в 2003 году (Женева) и в 2005 году (Тунис). Саммит обнажил недовольство текущим положением дел со стороны некоторых 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.

Ключевым являлся вопрос, что включает понятие «Управление Интернетом» и кто и как в нем может участвовать. Здесь столкнулись две точки зрения.

Первая, практическая, исходила из того, что Интернет в значительной степени — это результат совместной работы частного сектора, и он во многом обязан своему успеху именно отсутствию регулирования и «советов» правительств. Более радикальная версия этой позиции была провозглашена в 1996 году Джоном Барлоу (John Perry Barlow) в «Декларации независимости киберпространства»[41].

Вторая, традиционная, отстаивала принцип, что принятие решений как на национальном, так и международном уровнях является прерогативой государства и основой демократического волеизъявления.

Конечно, в рамках WSIS примирить такие позиции было практически невозможно.

Самое существенное, о чем удалось договориться, — это «определение» управления использованием Интернета в рамках Тунисской программы для информационного общества. Оно звучит следующим образом: «Рабочее опре-

[40] В новом уставе ICANN предусматривалось, что CAG может консультировать Совет ICANN по вопросам политики. В случае, если совет решит не прислушиваться к рекомендациям, обе стороны постараются найти компромиссное решение. Если же совет по-прежнему настаивает на своей позиции, он может ее принять, но должен объяснить причины своего выбора

[41] A Declaration of the Independence of Cyberspace, John Perry Barlow, https://www.eff.org/cyberspace-independence

деление управления использованием Интернета означает разработку и применение правительствами, частным сектором и гражданским обществом в рамках исполнения ими своих соответствующих ролей общих принципов, норм, правил, процедур принятия решений и программ, которые формируют условия для развития и использования Интернета[42]».

Другим достижением явилась договоренность о проведении 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.

Подытоживая этот раздел, нужно заметить, что вопросы «управления Интернетом» далеки от однозначного решения. Во многом потому, что не совсем понятно: чем конкретно управлять? Правительства с трудом решают эти вопросы на национальном уровне, что уж говорить о выработке приемлемой глобальной позиции, выходящей за рамки общих принципов!

Наиболее конкретными являются три центральные функции IANA и ICANN. Из них регистрация параметров протоколов вряд ли привлекает внимание общественности — отчасти из-за узкотехнического характера, отчасти потому, что IANA по существу обслуживает регистратуру для протоколов IETF.

В отношении системы распределения адресных и номерных ресурсов делались попытки «захвата»: например, МСЭ выдвинул предложение о создании новой регистратуры, которая производила бы распределение адресов странам, а не сетям. Однако опустошение пула адресов IPv4 сделало эту идею гораздо менее привлекательной. Самой заметной функцией IANA и ICANN по-прежнему является координация корневого уровня DNS. Попытки построения более сбалансированной системы принятия решений в этой области во многом определили и продолжают определять ландшафт Internet Governance.

Начиная с 2020 года в рамках ООН началась работа по подготовке так называемого Глобального цифрового договора (Global Digital Compact). Согласно сайту ООН «ожидается, что Глобальный цифровой договор «обозначит общие принципы открытого, свободного и безопасного цифрового будущего для

[42] Tunis Agenda for the Information Society, https://digitallibrary.un.org/record/565827?ln=en

всех»»[43]. Вопросы, которые этот договор может охватить, включают связность, недопущение фрагментации Интернета, больший контроль индивидуумами использования своих персональных данных, права человека в Интернете и продвижение заслуживающего доверия Интернета путем введения критериев ответственности за дискриминацию и распространение ложного контента. Также предполагается, что Договор позволить укрепить систему многостороннего управления Интернетом. И хотя принцип «мультистейкхолдеризма» упоминается достаточно часто, роль неправительственных организаций в принятии решений не выглядит значительной, даже в качестве консультативной. Возможно, это признак взросления Интернета, который стал частью цифровой и телекоммуникационной индустрии, где национальное регулирование и международные договоры являются основными инструментами управления.

В 2025 году состоится подведение итогов 20-летия Тунисской программы. Сохранит ли Интернет свой уникальный статус и свойства – время покажет.

Заключение

Не существует схемы строения Интернета, отсутствуют и точные данные о его топологии. Сеть развивалась эволюционно, бесконтрольно, благодаря независимому взаимодействию ее компонентов — отдельных сетей, поставщиков услуг, провайдеров контента и, конечно, пользователей. Эта естественная эволюция превратила Интернет из технологической платформы в экосистему — со своими законами и подчас невидимыми взаимозависимостями между ее различными частями.

Основу технических «законов» составляют стандарты Интернета, большинство из которых разработаны в IETF и W3C. Существуют также принципы, рекомендующие архитектурные решения и характер взаимодействия между отдельными компонентами. Сюда можно отнести и принцип прозрачности, и принцип сетевой нейтральности. Многие из них лишь частично находятся в технической плоскости и во многом затрагивают экономические аспекты и области регулирования. Отдельный раздел свода законов Интернета образуют политики и правила, с которыми мы познакомились, говоря о системе адресации, имен и разработки стандартов. Ну и конечно, существуют реальные законы, регулирующие деятельность в Интернете.

Помимо законов и принципов, развитие Сети во многом определяют экономические факторы. Независимые участники принимают решения и, взаимодействуя с другими, находят взаимовыгодные компромиссы. Взаимозависимости между компонентами Сети существуют благодаря глобальной связности,

[43] Глобальный цифровой договор, https://www.un.org/techenvoy/ru/global-digital-compact

но их реальный характер, а также последствия одностороннего изменения такой зависимости зачастую трудно предположить.

Не существует генерального плана развития Сети. Скоро ли наступит переломный момент во внедрении IPv6 и сохранит ли уровень Интернета статус «универсального коннектора»? Или, подобно тому, как Интернет в свое время был построен на базе отживающих технологий телефонных сетей, на устаревающем фундаменте IP возникнет новая архитектура Сети, где центральным элементом станут данные, а не каналы, узлы и связность? Возникнут ли новые бизнес-модели — или мы, пользователи, и наши персональные данные будут целиком поглощены поставщиками контента и связанными с ними рекламодателями? Возможно ли эффективно решить вопросы кибербезопасности, защиты личных данных и интеллектуальной собственности без существенного изменения фундаментальных качеств Сети, таких как прозрачность, глобальная связность, открытость и доступность?

Предсказать будущее Интернета трудно, практически невозможно. В этом суть генерирующей платформы: с виду неприметные ростки могут завтра удивить нас неожиданными всходами, тогда как многомиллионные инвестиции оставят только след на бумаге. Точно ответить на эти вопросы поможет только время.


КЭШ
КОНТЕНТ
СЕНСОР
BLUETOOTH
6 LOWPAN
RFC4944
IEEE 802.15.4
RFC6282
RFC6775
RFC7540
RFC7428
THREAD
ZIGBEE
LORA
NEST
HTTPS
AKAMAI
SIGFOX
Z-WAVE
ХАБ
HUE
IOT
DNS
CDN
Z-WANE
LIMELIGHT NETWORKS
ЭНЕРГОЭФФЕКТИВНЫЕ СЕТИ
БЕСПРОВОДНАЯ СВЯЗЬ
ИНТЕРНЕТ ВЕЩЕЙ
ГРАНИЧНЫЙ УЗЕЛ
УМНЫЕ ОБЪЕКТЫ
ГЛАВА 5

Глава 5

Будущее начинается сегодня

> Мы должны признать, что успешная архитектура имеет свойство развиваться, поскольку успех ведет к росту, а технология движется вперед. Вследствие этого различные допущения должны периодически пересматриваться при обновлении протоколов.
>
> *RFC 6250[1], май 2011 г.*

В предыдущих главах мы говорили об эволюции Интернета как платформы, где инновации происходят на различных уровнях в различные временные периоды. Мы также говорили об оссификации, или окаменелости, некоторых уровней и приложений — в первую очередь на уровне IP.

Сегодня мы наблюдаем наиболее бурное развитие именно на уровне приложений и связанного с ними контента. И хотя сетевой и транспортный уровни Интернета выполняют основную функцию обеспечения глобальной связности, новые приложения требуют от коммуникационной среды параметров, которые эти уровни напрямую не в состоянии предоставить.

Однако это не является проблемой, а скорее показывает работоспособность модели Интернета — низшие уровни не перегружаются дополнительной функциональностью, которая может быть реализована на более высоких уровнях.

В этой главе мы рассмотрим два примера. Они иллюстрируют, как на базе фундаментальной инфраструктуры могут быть построены целые экосистемы с параметрами,

[1] RFC 6250: Evolution of the IP Model, URL: https://www.rfc-editor.org/rfc/rfc6250

которых немыслимо ожидать от «простого» Интернета. В качестве первого примера возьмем сети доставки контента — CDN. Здесь мы посмотрим, как параметры высокой масштабируемости, надежности и качества могут быть достигнуты путем создания оверлейных систем. Второй пример расскажет об Интернете вещей, или IoT (Internet of Things). В случае IoT мы являемся свидетелями эволюции, когда обыденные вещи начинают видеть, думать и действовать. Мы наблюдаем внедрение проникающих систем управления и контроля — в индустрии, офисе и дома. Это расширяет наши возможности, но таит и существенные риски.

О новых технологиях, протоколах и системах, которые позволяют нам лучше представить, что ждет нас завтра, — в этой главе.

Что такое сети доставки контента и зачем они нужны?

Как мы обсуждали в предыдущих главах, основной задачей опорной инфраструктуры Интернета является пересылка пакетов от одного узла к другому в надежде, что пакет достигнет своего места назначения в целости и в срок. Но Сеть сама никаких гарантий на этот счет не дает. Если пакет окажется утерянным вследствие перегрузки какого-либо участка Сети или будет доставлен с опозданием, вне изначального порядка (например, вследствие недоступности основного маршрута), — то приложениям придется обрабатывать такие нерегулярности соответствующим образом. Пропускная полоса коммуникационного канала между приложениями также не гарантирована и заранее не известна. Этот параметр определяется множеством факторов, начиная с пропускной способности отдельных сегментов сетей, через которые передается трафик между приложениями, и заканчивая загрузкой самих сетей.

В начале 90-х годов прошлого столетия большинство приложений Интернета были весьма «эластичными», а именно малочувствительными к параметрам передачи. Например, обмен файлами с помощью протокола FTP или отображение веб-страницы могли занять секунды, минуты или часы, но фундаментального значения для приложения это не имело. Новые же приложения — приложения реального времени, интерактивные приложения и просто современные веб-порталы — потребовали определенных стабильных параметров качества.

В главе 3 мы говорили о попытках разработать и внедрить решения по «повышению качества» в Интернете. Эти решения были стандартизованы, однако их внедрение потерпело полное фиаско. Во многом потому, что они требовали удорожания и усложнения опорной инфраструктуры, пересмотра существующих взаимоотношений по обмену трафиком между операторами, но также и потому, что в конце 1990-х значительно упала стоимость пользования каналами передачи данных, в том числе и международными. Соответственно, возросла и скорость доступа для конечного пользователя.

Тем не менее случаи, когда для одного пользователя портал онлайн-магазина загружается за две секунды, а для другого — за несколько минут, являлись вполне типичными. Еще пример: после минуты нормального воспроизведения видеоролик внезапно замирает, пусть и на секунды. Такие ситуации, конечно, нельзя назвать приемлемыми.

Но Интернет — это сложная система взаимодействующих сетей, которые управляются независимо и различаются топологически и технологически. Возьмите хотя бы такой простой фактор, как расстояние. В таблице 1 показано влияние расстояния между клиентом (веб-браузером) и сервером на пропускную способность виртуального канала стандартного TCP.

Таблица 7. Влияние расстояния на пропускную способность

Масштаб расстояния	Время RTT	Типичная потеря пакетов	Эффективная пропускная способность	Время скачивания 4ГБ DVD
Местный (<160 км)	1,6 мс	0,6%	44 Мб/с (высококачественный поток HDTV)	12 мин
Региональный (900-1800 км)	16 мс	0,7%	4 Мб/с (минимальный поток HDTV)	2,2 часа
Межконтинентальный 5000 км	48 мс	1,0%	1 Мб/с (поток телевидения стандартного качества)	8,2 часа
Многоконтинентальный 10 000 км	96 мс	1,4%	0,4 Мб/с (видеопоток слабого качества)	20 часов

Источник: Erik Nygren, Ramesh Sitaraman and Jennifer Sun
The Akamai Network: A Platform for High-Performance Internet Applications
ACM SIGOPS Operating Systems Review Vol. 44 Iss. 3 (2010)

Перефразируя известную поговорку, если пользователь не может забрать необходимый контент, контент должен быть доставлен пользователю! Так появилась идея сетей доставки контента — CDN (Content Delivery Networks) — надстройки или еще одного приложения Интернета.

Краткая история CDN

Появлению первых CDN в конце 1990-х предшествовало внедрение таких технологий, как серверные фермы с балансировкой загрузки, иерархические кеши и веб-прокси. Они позволяли улучшить производительность веб-сервера и доставку контента. В Массачусетском технологическом институте (MIT) Том Лейтон (Tom Leighton) и Данни Левин (Danny Lewin) разработали математические алгоритмы для оптимальной маршрутизации и репликации контента в широкомасштабной сети распределенных серверов. Эти алгоритмы впоследствии легли в основу архитектуры крупнейшей CDN Akamai.

Внедрению и развитию первых CDN способствовало новое явление, получившее название «внезапной толпы» (Flash crowd), также называемое эффектом Slashdot. Суть его сводится к тому, что на малозаметный доселе веб-сайт вдруг обрушивается шквал популярности, который делает его практически недоступным вследствие перегрузки как самого сайта, так и сетевой инфраструктуры. Так, многие новостные сайты не выдержали нагрузки после известных событий 9/11 в США.

Первое название, Flash crowd, произошло от одноименного фантастического романа Ларри Нивена (Larry Niven), написанного в 1973 году. По его сюжету, была изобретена телепортация, и люди смогли свободно и мгновенно переноситься в те места, где происходит что-нибудь любопытное. Второе название происходит от новостного сайта Slashdot[2], публикация дайджеста в котором со ссылкой на оригинальный сайт иногда давала эффект, схожий с атакой отказа в обслуживании из-за наплыва посетителей.

В 1999 году Akamai запустила коммерческую службу доставки контента, клиентом которой стал один из наиболее посещаемых сайтов — Yahoo!

По мере роста потребности в качественной доставке различных видов контента другие компании также занялись построением CDN. В качестве наиболее известных примеров можно привести Limelight Networks (2001), EdgeCast (2006), Amazon (2008), CloudFlare (2009), Incapsula (2009).

Первое поколение CDN было направлено на улучшение производительности веб-сайтов и обслуживало преимущественно статический контент. Основным подходом к построению таких сетей являлось размещение кеширующих серверов (серверных ферм) в максимальной близости от провайдеров доступа, обслуживающих конечных пользователей. Наиболее типичным являлось обеспечение присутствия в точках обмена трафиком.

С появлением услуг видео по требованию, обеспечивающих потоковую передачу, встала необходимость в архитектурном подходе, отличном от традиционных CDN. Дело в том, что видеоклип по существу мало отличается от большого файла, но его передача должна быть синхронизирована с потоковым сервером, воспроизведение может быть начато и прервано в произвольный момент.

Архитектура CDN

Идея CDN проста: разместить реплики контента как можно ближе к его потенциальным потребителям и тем самым обеспечить стабильную необходимую пропускную способность, а также распределить нагрузку на серверы-поставщики контента, например, на веб-серверы. Реализация этой идеи в глобальном Интернете гораздо сложнее и требует ответа на вопросы: где и сколько реплик должно быть размещено, как синхронизировать контент, каким образом осуществлять перенаправление запросов пользователя к нужной реплике?

[2] https://slashdot.org

Решение связанных с этим задач требует создания распределенной наложенной сети с функциями управления и мониторинга. Можно выделить следующие основные компоненты CDN, представленные на рис. 79.

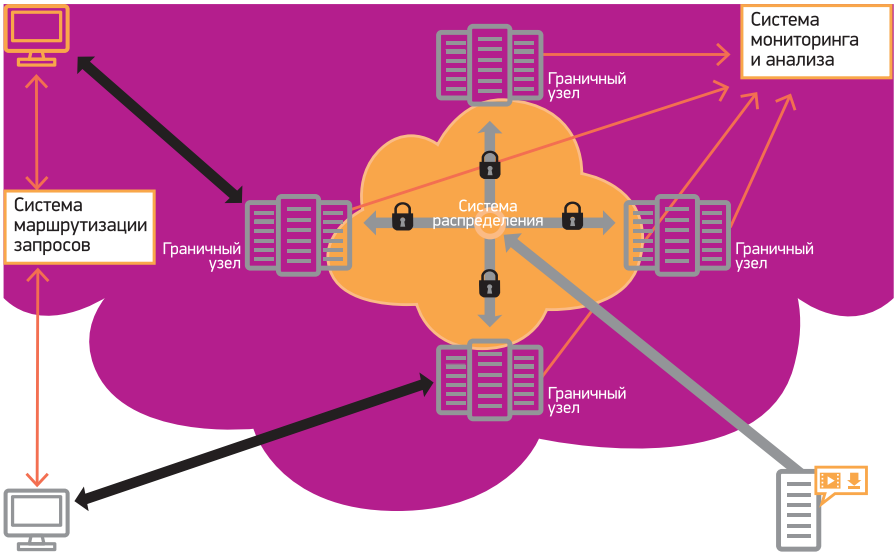

Рис. 79. Основные компоненты CDN.

Граничные кеширующие узлы

Кеширующие узлы, объединенные в наложенную или выделенную сеть, обеспечивают доставку контента на «последней миле». От того, насколько географически распространена сеть и насколько «правильно» расположены граничные узлы, зависит общая производительность и качество CDN.

Граничные узлы могут представлять собой компьютерные кластеры для обеспечения надежности и распределения нагрузки. Обычно они размещаются вблизи точек обмена трафиком, где можно обеспечить связность CDN со множеством сетевых операторов.

Многие крупные CDN, такие как Akamai, Google, Netflix и Facebook, используют так называемые внутрисетевые (on-net) кеши внутри сетей доступа, которые взаимодействуют с пограничными серверами и доставляют контент еще ближе к конечному пользователю. С распространением сетей 5G и облачных услуг на основе граничных вычислений с множественным доступом (Multi-access edge computing, MEC) такие сетевые кеши могут располагаться еще ближе к пользователю — на границе базовой сети оператора мобильной связи. Распределение контента в эти локальные кеши является многоуровневым, что еще существеннее минимизирует дублирование трафика и задержки.

Запросы клиентов обслуживаются самими узлами локально, если запрашиваемые данные находятся в кеше. При их отсутствии — данные копируются в кеш от исходного сервера. Однако кеширующие узлы отличаются от стандартного прокси, поскольку могут выполнять ряд других функций, перечисленных ниже.

1. **Управление кешем**
   Разные классы контента требуют различных характеристик кеша. На основе статистической популярности и метаданных контента (например, заголовков Last-Modified) узел может вычислить возможное время истечения годности кеша и заранее освежить данные от источника. Это особенно важно в случае, когда исходный сервер не предоставляет информацию по управлению кешем (с помощью заголовков cache-control, etag, expires). Но даже если сервер использует заголовки для управления кешем, они рассчитаны на кеширование браузером и не всегда адекватно работают в условиях CDN.

   Поскольку CDN располагает значительной информацией относительно конкретного контента — его типа, характера запросов и т.п., управление кешем может обеспечиваться более динамично, нежели с помощью заданных статических параметров кеша на сервере. Например, даже динамический контент, который обычно не подлежит кешированию, имеет определенный период годности, в течение которого граничный узел может отвечать на запросы клиентов без обращения к исходному серверу.

2. **Определение местонахождения источника контента, включая обработку ситуаций недоступности**
   С помощью системы маршрутизации запросов, о которой мы поговорим далее, узел может принять решение о выборе оптимального экземпляра исходного сервера, модификации запроса (модификации URL) или перенаправлении запроса на другой граничный узел.

3. **Изменение HTTP-заголовков**
   Граничные узлы могут добавлять, удалять или изменять заголовки HTTP-запросов и ответов, особенно заголовки куки (cookie), такие как set-cookie и cookie. Это может использоваться для обмена служебной информацией с сервером-источником и с пользователями, а также для управления каскадными кешами.

Транспортная система доставки контента

В задачу транспортной системы входит оптимальная доставка требуемого контента к граничным кеширующим узлам. Небольшие CDN могут себе позволить одноуровневую структуру, когда граничные узлы посылают запросы на обновление кеша непосредственно веб-серверам. При этом для связности часто используется Интернет, а не выделенная сеть.

Крупные CDN используют многоуровневую структуру, когда группы граничных узлов приписаны к кеширующим кластерам более высокого уровня. Таким образом уменьшается нагрузка на веб-серверы — источники контента за счет увеличения частоты попадания в кеш (cache hit), а также уменьшается задержка доставки за счет дополнительного уровня распределения.

В качестве транспортной сети в крупных CDN используется либо наложенная, либо выделенная сеть. Во многих случаях применяется оптимизация маршрутизации с использованием непрерывных тестов связности и пропускной способности между узлами. Все чаще для управления передачей трафика в транспортной сети используется архитектура SDN, о которой мы говорили в главе 3 «Глобальная система маршрутизации и передачи данных».

Более сложная структура иерархических кешей используется при необходимости доставки видеопотоков, особенно в реальном времени. От граничных узлов также требуется дополнительная функциональность, так как они играют роль потоковых серверов. Потоковые серверы обеспечивают необходимую фрагментацию потока, его кодирование с различным разрешением/качеством и взаимодействие с клиентами.

Например, в сети Akamai в качестве узлов транспортной сети используется система «входных узлов» (entrypoint), расположенных в непосредственной близости к поставщику контента, и рефлекторов (set reflector), обеспечивающих оптимальную доставку данных в граничные узлы.

Система маршрутизации запросов

«Мозгом» CDN является система маршрутизации запросов. В задачи этой системы входит выбор оптимального граничного узла для обслуживания запроса клиента. Выбор осуществляется на основе нескольких параметров, включая близость узла к клиенту, загрузку узла, наличие в кеше требуемого контента, а также других возможных правил (например, связанных с защитой авторских прав или доступности услуг в определенных географических областях). Для достижения этой цели используется либо DNS, либо Anycast BGP. Далее мы рассмотрим оба метода более подробно.

Работа CDN

При отсутствии CDN, когда пользователь пытается получить доступ к онлайн-контенту, например, к веб-серверу, он направляется на исходный сервер. В простой настройке имя хоста в URL-адресе, например, https://www.example.com, преобразуется DNS в IP-адрес исходного сервера (93.184.216.34). Как только соединение HTTP(S) установлено, содержимое доставляется пользователю.

Когда веб-сервер использует CDN для доставки контента, пользователь должен быть направлен на оптимальный граничный узел, а не на исходный сервер. Эту задачу решает система маршрутизации запросов.

Когда пользователь запрашивает контент с граничного узла, а контент там уже существует, он будет предоставлен немедленно. Это называется «кеш-попадание» (cache hit). Если запрошенный контент просрочен в кеше пограничного сервера или никогда не кешировался, граничный узел должен будет сначала загрузить его с исходного сервера. Это называется «промах кеша» (cache miss). Хотя это может привести к некоторому снижению производительности, обычно контент, связанный с запрашиваемым, также подгружается, поэтому дальнейшие запросы с большой вероятностью будут попадать в кеш.

Маршрутизация запросов с использованием DNS

Этот метод является наиболее популярным вследствие повсеместного использования DNS. В его основе лежит применение специализированного DNS-сервера для трансляции имени ресурса, например, веб-сервера. В зависимости от указанных выше параметров выбора (например, близость узла к клиенту, загрузка узла, наличие в кеше требуемого контента) сервер возвращает различные записи ресурсов A/AAAA, CNAME или NS. Об этих записях мы подробнее говорили в главе 2 «Глобальная система имен».

Так, например, в простейшем случае сервер может вернуть один или несколько IP-адресов оптимального граничного узла. Несколько IP-адресов позволяют осуществлять простейшую балансировку загрузки, когда клиенты будут получать поочередно IP-адреса из множества, тем самым распределяя нагрузку среди нескольких граничных узлов.

В более сложных ситуациях может использоваться многоуровневая трансляция имени, позволяющая распределить процесс принятия решений по маршрутизации запроса между основным DNS-сервером и несколькими специализированными и обладающими более полной информацией DNS-серверами. Для этого используются записи NS или CNAME.

Основной недостаток использования записи NS для «каскадирования» процесса трансляции — то, что число уровней ограничено числом частей самого имени. Например, имя a.b.example.com допускает лишь один уровень «каскадирования» от сервера, отвечающего за example.com, к серверу, авторитетному в зоне b.example.com, который, в свою очередь, вернет IP-адрес a.b.example.com. Поэтому наиболее распространенным является использование CNAME, которое не накладывает такого ограничения.

Например, для получения IP-адреса граничного узла CDN Akamai, обслуживающего сервер www.apple.com, клиенту необходимо обработать три перенаправления: сначала на www.apple.com.edgekey.net., затем на www.apple.com.edgekey.net.globalredir.akadns.net. и, наконец, на e6858.dscc.akamaiedge.net., адрес которого укажет на оптимальный для клиента граничный узел. Этот процесс схематично представлен на рис. 80.

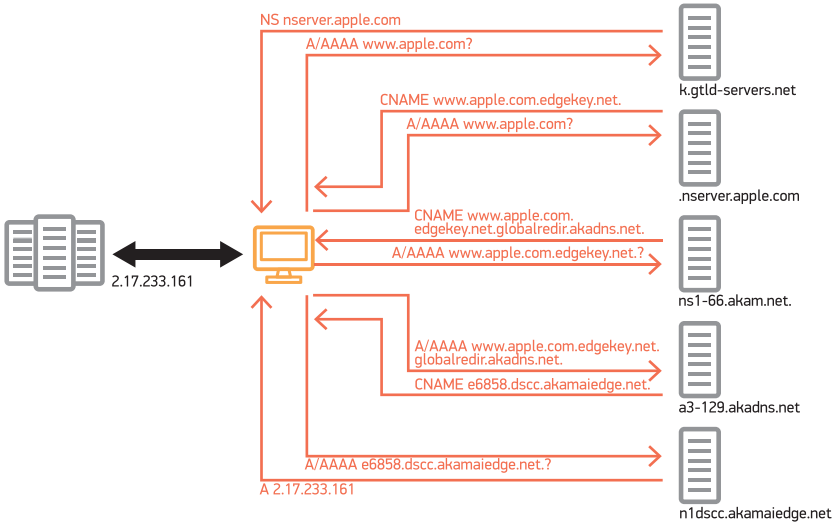

Рис. 80. Каскадная трансляция имени www.apple.com для определения оптимального граничного узла. (Схема указывает принцип работы, так что конкретные имена серверов имен и IP-адреса могут отличаться.)

Как мы видим, DNS-серверы, осуществляющие эти перенаправления, на самом деле являются компонентами системы маршрутизации запросов и генерируют имена в соответствии с алгоритмом поиска оптимального граничного кеширующего узла.

Однако использование DNS для маршрутизации запросов имеет существенные ограничения.

Во-первых, маршрутизация происходит на уровне запрашиваемого доменного имени, хотя часто предпочтительнее была бы маршрутизация на уровне запрашиваемого объекта. Например, объекты контента с различными характеристиками (размер, частота обновления, транспортные характеристики) могут быть размещены на различных кеширующих узлах. И в то же время все они могут быть адресованы различными URL с общим доменным именем. Во-вторых, для определения местонахождения клиента используется IP-адрес DNS-резолвера клиента, а не самого клиента, поскольку DNS-запрос на трансляцию имени поступает именно от резолвера. В некоторых случаях это может привести к существенным ошибкам в определении «ближайшего» граничного узла. Особенно — когда используются резолверы, расположенные вне сети клиента, например, общедоступные резолверы PublicDNS или OpenDNS. Для решения этой проблемы в IETF недавно был документирован механизм[3], позволяющий резолверу при обращении к авторитетному DNS-

[3] RFC 7871 Client Subnet in DNS Queries, https://www.rfc-editor.org/rfc/rfc7871

серверу указать префикс сети клиента (обычно /24 в случае IPv4 и /56 для IPv6). Для передачи этой информации используется специальная опция механизма расширений DNS EDNS0. Однако многие общедоступные резолверы за исключением, пожалуй, PublicDNS (Google) и OpenDNS (Cisco), не поддерживают эту опцию. Отчасти это связано с нежеланием раскрывать информацию ввиду ее частного характера, отчасти с тем, что многие общедоступные резолверы используют глобально распределенную сеть узлов, обеспечивающих эту услугу с помощью технологии anycast. Поэтому запросы приходят от топологически ближайшего к пользователю узла.

Тем не менее данная проблема отчасти остается актуальной, как, впрочем, и проблема недостаточной детализации конкретного запрашиваемого объекта. Для ее решения используются дополнительные методы маршрутизации на транспортном уровне и уровне приложений.

Маршрутизация запросов на транспортном уровне

При использовании этого механизма система маршрутизации анализирует информацию первого пакета запроса клиента на транспортном уровне. Обычно анализом пакета занимается граничный узел, который был выбран с использованием DNS. При этом система получает информацию о фактическом IP-адресе клиента и протоколе приложений, который этот запрос использует. Эта информация позволяет определить оптимальный граничный узел и при необходимости перенаправить запрос.

Маршрутизация запросов на уровне приложений

Анализ запроса на уровне приложений дополнительно позволяет получить информацию о запрашиваемом объекте (или объектах) контента и, соответственно, обеспечить оптимальную маршрутизацию.

Типичным является анализ запрашиваемого URL для более точного определения обслуживающего граничного узла или ближайших кешей более высокого уровня, содержащих запрашиваемые данные. Также часто используется анализ заголовков, таких как Cookie, Accept-language и User-agent, для определения оптимального источника данных. Cookies применяются для идентификации клиента сайта или веб-сессии.

Для перенаправления запроса используется возможность протокола HTTP сигнализировать новый маршрут с помощью кода перенаправления (коды 302 Found или 307 Temporary Redirect). В этом случае ответ содержит новый URL, который приложение должно использовать для доступа к контенту.

Иногда маршрутизация запросов для встроенных объектов, например, изображений, может осуществляться в момент загрузки страницы. В этом случае используется техника «переписывания URL», когда встроенные директивы HTTP переписываются на лету, указывая на оптимальное для данного клиента расположение объектов.

Anycast CDN

Другой подход к определению пограничного сервера, ближайшего к пользователю, состоит в том, чтобы оставить это решение системе интернет-маршрутизации, используя технологию anycast. В соответствии с RFC4786[4] anycast основана на анонсировании IP-адреса сервера (и соответствующего сервиса) в нескольких топологически распределенных местах, так что отправленные дейтаграммы направляются в одно из нескольких доступных мест в соответствии с решениями, принятыми системой маршрутизации.

При использовании anycast всем граничным узлам назначается один и тот же набор IP-адресов, которые анонсируются в системе маршрутизации в местах их конкретного расположения. В этом случае трафик направляется на «ближайший» граничный узел с точки зрения системы маршрутизации, что обычно означает минимальное количество промежуточных сетей (автономных систем в терминах BGP) между пользователем и граничным узлом.

Преимуществом этого подхода является его простота. При достаточном распределении граничных узлов в регионе обслуживания CDN может быть достигнута достаточная близость «ближайшего узла» к пользователю. Одним из основных недостатков anycast является то, что выбор граничного узла в основном находится вне контроля CDN и зависит от топологии межсетевых соединений и политик маршрутизации сетей, участвующих в передаче трафика. Из-за этого управление распределением нагрузки на граничные узлы и другими параметрами, такими как задержка, может быть затруднено.

Google Cloud CDN — это пример CDN, использующей anycast.

P2P CDN

Одной из проблем CDN является плотность покрытия и «последняя миля». Даже при наличии собственной магистрали и тысяч граничных узлов существует практический предел плотности покрытия. Последняя миля, оконечный сетевой сегмент провайдеров широкополосного доступа, в большинстве случаев находится вне контроля CDN.

Для решения этой проблемы иногда используется одноранговый (peer-to-peer, или P2P) подход к доставке контента, т.н. P2P CDN. Он не требует выделенной инфраструктуры и полагается на то, что пользовательские устройства получают доступ к одному и тому же контенту и обмениваются им друг с другом вместо того, чтобы получать его с граничного узла или исходного сервера.

P2P CDN в основном используется для потоковой передачи контента и доставки больших файлов, таких как игровое приложение или обновление ОС. Хотя эту технологию можно использовать для построения автономных CDN, она часто

[4] RFC4786: «Operation of Anycast Services», URL: https://www.rfc-editor.org/rfc/rfc4786

используется в качестве гибридного подхода, когда традиционная CDN доставляет контент на граничный узел, а P2P распределяет его среди потребителей/пользователей сети. Технология, обычно используемая для обмена данными между узлами в P2P CDN, —это WebRTC[5].

Обслуживание статического и динамического контента

Производительность CDN сильно зависит от того, насколько «кешируемым» является исходный контент. Действительно, разница в производительности между попаданием в кеш и промахом может быть значительной. В этом отношении важно понять, как CDN работает со статическим и динамическим контентом.

Статическое содержимое — это содержимое, которое всегда остается одним и тем же, даже если к нему обращаются разные пользователи. Каскадные таблицы стилей CSS, которые применяются для оформления сайта, JavaScript для обеспечения интерактивности, видео и изображения — все это, как правило, не меняется для каждого пользователя для данного URL-адреса и, таким образом, полностью кешируется.

Динамический контент — это контент, который генерируется для пользователей «на лету», когда они просматривают веб-страницу. Динамический контент может быть намного более ресурсоемким, поскольку он может выполнять код и обычно требует запросов к базе данных для создания контента. Многие системы управления контентом (CMS), такие как WordPress, или персональные порталы в области, например, банковского дела или здравоохранения, генерируют динамический контент, который может не полностью кешироваться или вообще не кешироваться.

Традиционно CDN обслуживали статический контент. Запросы динамических элементов, по сути, означали промахи кеша и должны были перенаправляться на исходный сервер. Однако новые методы, такие как интеграция с популярными CMS через API и заголовки управления HTTP, позволяют более эффективно кешировать динамический контент.

Хотя видеопоток выглядит как динамический контент, на самом деле он статичен. Стандартные отраслевые протоколы потоковой передачи, такие как HLS (HTTP Live Streaming)[6] и MPEG-DASH[7], основаны на концепции сегментации. Видеоклип нарезается на отдельные куски или сегменты, каждый из которых длится несколько секунд. Каждый из сегментов имеет отдельный URL и поэтому может кешироваться независимо. Видеопроигрыватель пользователя использует файл «манифеста», связанный с видео, в котором перечислены URL-адреса для всех сегментов видеоклипа. Видеопроигрыватель отправляет HTTP-запросы

5 Real-time communication for the web, URL: https://webrtc.org/
6 https://ru.wikipedia.org/wiki/HLS
7 https://ru.wikipedia.org/wiki/MPEG-DASH

на получение этих сегментов, которые обслуживаются либо из кеша граничного узла (в случае совпадения), либо сначала загружаются с исходного сервера (в случае ошибки).

Инфраструктура с несколькими CDN

Хотя использование CDN улучшает доставку контента по сравнению с одним веб-сайтом, есть несколько соображений, которые поставщик контента должен учитывать при выборе CDN. Ценообразование, дополнительные услуги и общая производительность входят в их число. Но не менее важны географический охват и соображения устойчивости.

Например, одна CDN может обеспечивать превосходное покрытие в отдельных регионах, в то время как другие территории остаются без покрытия. В то же время другая CDN может быть ориентирована именно на эти места, и ее сервис будет дополнять основной. В другом примере провайдер хочет снизить вероятность сбоя, вызванного CDN, и создать план резервного копирования. В обоих случаях использование нескольких CDN вместо одной может стать оптимальным решением.

Типичный способ реализации настройки с несколькими CDN выглядит следующим образом. Предположим, поставщик контента планирует использовать две CDN. Тогда провайдер может использовать DNS в качестве «диспетчера» между ними. Если провайдер поддерживает DNS для своего домена, он может направить разрешение имен на соответствующие серверы имен CDN в циклическом режиме (именно так работает DNS, если присутствует несколько записей для одного и того же имени). Однако такой подход не будет учитывать географическое распределение.

Существуют специализированные провайдеры DNS, которые предлагают услугу «диспетчеризации» на основе геолокации пользователя и даже некоторых общих показателей производительности используемых CDN. Например, компания NS1[8] предлагает интеллектуальные службы DNS, которые направляют пользователя на CDN, оптимальную в данных обстоятельствах (местоположение, производительность и т.д.). Контент-провайдеру нужно просто направить разрешение имен на NS1 или полностью отдать DNS на аутсорсинг для выполнения этой задачи.

Некоторые CDN предлагают использование дополнительной CDN в качестве услуги. Например, клиенты Akamai могут настроить другого провайдера CDN для использования в определенном регионе, при определенной производительности, нагрузке и т.д.

8 https://ns1.com

Взаимодействие CDN

Эффективность CDN во многом определяется тем, насколько близко точки ее присутствия находятся к потребителю контента. Поэтому многие крупные CDN насчитывают десятки тысяч граничных серверов с глобальным покрытием. Например, на август 2016 года CDN компании Akamai насчитывала более 216 000 серверов, расположенных в 120 странах и работающих внутри более чем 1500 сетей. Но даже такая впечатляющая сеть не может покрыть весь Интернет. В то же время зоны покрытия существующих CDN имеют значительные перекрывающиеся области. Наконец, относительно небольшие CDN, чаще всего развернутые в рамках провайдера интернет-доступа или корпоративной сети, не могут достичь высокой эффективности в силу ограниченного покрытия.

Отметим, что в Интернете глобальная связность обеспечивается путем взаимодействия множества независимых сетей, а не одним суперпровайдером. Представим модель, в рамках которой аналогично происходила бы доставка контента. В рамках этой модели одни CDN могли бы осуществлять «делегирование» доставки другим CDN, тем самым максимизируя общую эффективность.

В этом случае возможным сценарием доставки контента становится следующий процесс:

- Изначальный запрос от потребителя контента (пользователя) принимается авторитетной CDN — то есть CDN, обслуживающей провайдера контента в рамках соответствующего соглашения. Часто такую CDN называют «CDN вверх по потоку» (Upstream CDN, uCDN).
- Авторитетная CDN может обслужить запрос самостоятельно или же перенаправить его другой CDN в случае, если последняя способна выполнить задачу лучше (например, вследствие близости к пользователю). Такую CDN называют «CDN вниз по потоку» (Downstream CDN, dCDN).
- В ответ браузер пользователя запросит контент у dCDN, который, если требуется, будет подкачан из авторитетной uCDN и, если необходимо, от провайдера контента.

Другой пример эффективного взаимодействия CDN — создание федераций. Многие крупные провайдеры широкополосного доступа внедряют собственные CDN. Однако поскольку зона обслуживания таких провайдеров часто включает различные регионы, соответственно, и CDN являются разобщенными. Обмен данными между ними путем создания «федеративной» CDN позволило бы решить эту проблему.

Рабочая группа IETF CDNI (Content Delivery Networks Interconnection) занимается документированием практики и стандартизацией протоколов, необходимых для осуществления обмена данными между автономными CDN.

Общая модель взаимодействия между двумя CDN представлена на рис. 81. Как видно из рисунка, обмен данными осуществляется между соответствующими подсистемами CDN, которые мы обсуждали ранее в этой главе.

Кратко остановимся на интерфейсах этого взаимодействия:

- CDNI Control interface (CI). Управляющий интерфейс, обеспечивающий обмен данными между системами управления CDN. Этот интерфейс необходим для инициализации всех остальных интерфейсов. В этом смысле он также иногда называется Trigger interface.
- CDNI Logging interface (LI). Через этот интерфейс происходит обмен информацией об операциях CDN. Например, взаимодействующие CDN могут использовать его для мониторинга трафика в реальном времени либо асинхронно обмениваться данными для анализа работы и финансовых расчетов.
- Интерфейс маршрутизации запросов отвечает за операции, необходимые для того, чтобы определить, какая CDN (и, возможно, какой граничный сервер) будет обслуживать запрос пользователя. Этот интерфейс состоит из двух взаимосвязанных интерфейсов:

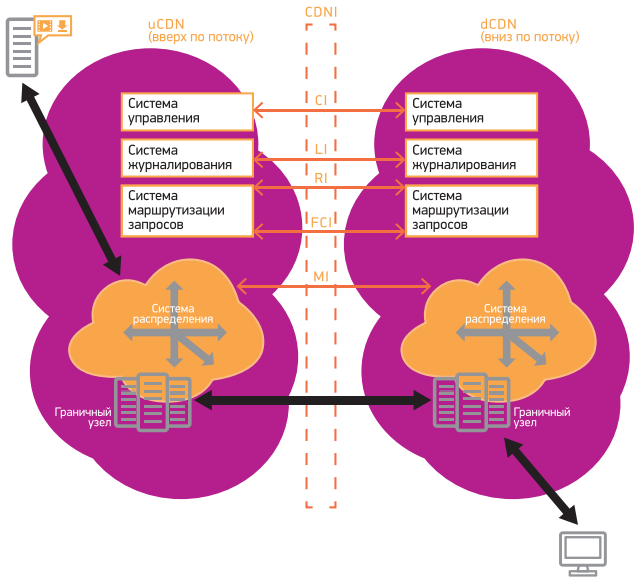

Рис. 81. Структура взаимодействия двух CDN в рамках модели CDNI.

1. CDNI Footprint & Capabilities Advertisement interface (FCI). Через этот интерфейс осуществляется асинхронный обмен информацией, требуемой для маршрутизации запросов, а именно информацией о зоне охвата и возможностях CDN;
2. CDNI Request Routing Redirection interface (RI). Этот интерфейс обеспечивает синхронные операции для определения CDN, отвечающей за доставку контента пользователю в ответ на конкретный запрос.

- CDNI Metadata interface (MI). Этот интерфейс обеспечивает обмен метаданными, определяющими параметры обслуживания контента CDN. В качестве примера таких метаданных можно привести указания по геоблокированию, временные промежутки доступности контента и параметры доступа.

Основные стандарты, определяющие параметры этих интерфейсов и формат данных:

CDN Interconnection Metadata[9]
Эта спецификация определяет формат метаданных и протокол для обмена. Метаданные, связанные с определенным контентом, предоставляют dCDN информацию, необходимую для обслуживания запросов пользователя.

Request Routing Redirection interface for CDN Interconnection[10]
Этот документ определяет формат данных и протокол запросов от uCDN к другой CDN о возможности последней обслужить конкретный пользовательский запрос. Также он определяет ответ обслуживающей dCDN, содержащий информацию о параметрах перенаправления пользовательского запроса.

CDNI Logging Interface[11]
Эта спецификация определяет структуру и протокол обмена информацией об операциях между взаимосвязанными CDN, а также формат файлов регистрационного журнала.

CDNI Control Interface / Triggers[12]
Эта спецификация определяет часть интерфейса CI, позволяющего одной CDN вызвать определенные действия со стороны другой CDN, которой делегирована доставка контента пользователю. Примером таких действий может быть подготовка контента и метаданных или очистка метаданных и кеша.

CDNI Request Routing: Footprint and Capabilities Semantics[13]
Этот документ определяет требования к протоколу FCI и, в частности, тип и формат данных, необходимых для извещения других CDN (выше по потоку) о возможностях и зоне охвата данной CDN.

---

9 RFC 8006: Content Delivery Network Interconnection (CDNI) Metadata, URL: https://www.rfc-editor.org/rfc/rfc8006

10 RFC 7975: Request Routing Redirection Interface for Content Delivery Network (CDN) Interconnection, URL: https://www.rfc-editor.org/rfc/rfc7975

11 RFC 7937: Content Distribution Network Interconnection (CDNI) Logging Interface, URL: https://www.rfc-editor.org/rfc/rfc7937

12 RFC 8007: Content Delivery Network Interconnection (CDNI) Control Interface / Triggers, URL: https://www.rfc-editor.org/rfc/rfc8007

13 RFC 8008: Content Delivery Network Interconnection (CDNI) Request Routing: Footprint and Capabilities Semantics, URL: https://www.rfc-editor.org/rfc/rfc8008

URI Signing for CDN Interconnection[14]
Этот документ описывает, как концепция электронной подписи URI обеспечивает контроль доступа, и предлагает метод подписи URI.

Вопросы безопасности

Вопросы безопасности, относящиеся к различным подсистемам Интернета, таким как DNS или маршрутизация, в полной мере применимы и к CDN. Однако для CDN существуют некоторые особые аспекты, которые мы рассмотрим ниже.

Влияние шифрования данных

Использование протокола TLS/HTTPS для шифрования данных и аутентификации веб-серверов становится все более общей практикой. При этом весь трафик между браузером и сервером зашифрован на уровне приложений. Таким образом, все данные HTTP, включая заголовки, URL и, собственно, сам контент, являются недоступными для промежуточных устройств.

Это, безусловно, представляет проблему для CDN. Отсутствие доступа к этим данным означает, что контент невозможно кешировать, не представляется возможным манипулировать заголовками и проводить анализ трафика для оптимальной маршрутизации запросов. Означает ли это, что сайты, использующие HTTPS, не могут эффективно обслуживаться CDN?

Поскольку соединения HTTPS завершаются на пограничных серверах, сеть CDN должна иметь как публичный сертификат TLS, так и закрытый ключ, соответствующий открытому ключу сертификата. Хотя владелец домена может сам получить сертификат TLS от доверенного удостоверяющего центра (УЦ) и передать его CDN, многие предлагают такую возможность в рамках пакета услуг. На самом деле, многие CDN являются УЦ, способными выдавать такие сертификаты.

Секретный ключ и сертификат TLS, объединяющего параметры ресурса (полное доменное имя и организацию-оператор) с соответствующим публичным ключом, размещаются на каждом граничном узле, обслуживающем определенный контент. Этот процесс схематично показан на рис. 82. Здесь каждый граничный узел может расшифровать передаваемые данные и вновь зашифровать их для передачи от поставщика контента (веб-сайта) и обратно по внутренней сети CDN. При этом могут использоваться другие ключи и схемы шифрования.

[14] RFC 9246: URI Signing for Content Delivery Network Interconnection (CDNI), URL: https://www.rfc-editor.org/rfc/rfc9246

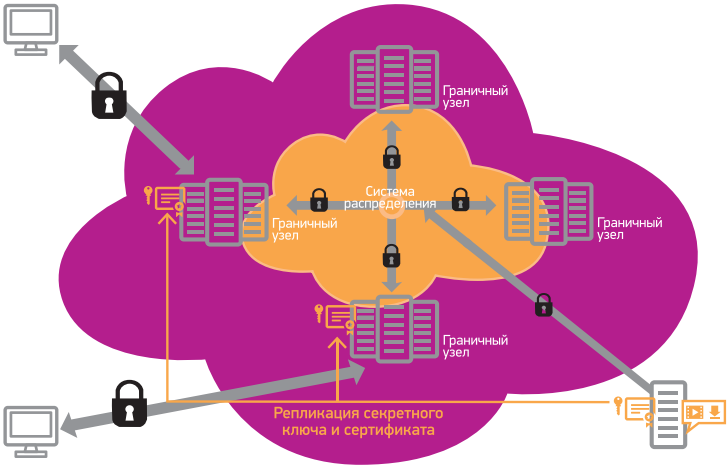

Рис. 82. Размещение TLS-сертификата и соответствующего секретного ключа на граничных узлах, обслуживающих зашифрованный контент.

Однако этот подход несет в себе существенные риски. Секретный ключ хранится в десятках, если не в сотнях тысяч узлов, каждый из которых может иметь уязвимые места, открывающие возможность несанкционированного доступа. Это представляет большую опасность и накладывает серьезные требования на защищенность узлов CDN. Добавим, что каждый граничный узел хранит не один, а множество секретных ключей сайтов, которые данная CDN обслуживает. В рамках совещания IETF96 в июле 2016 года было организовано обсуждение возможных решений данной проблемы на встрече BoF LURK (Limited Use of Remote Keys, Ограниченное использование удаленных ключей). Суть предлагаемой архитектуры заключалась в использовании специального сервера ключей для осуществления криптографических операций. В рамках этой схемы авторизованные граничные узлы обращаются к серверу ключей для шифрования/расшифровывания данных. Даже если один из граничных узлов скомпрометирован, это повлияет только на запросы, обслуживаемые данным узлом, а не на всю систему в целом, как в случае неавторизованного доступа к секретному ключу. Однако эти идеи не получили дальнейшего развития.

Противодействие распределенным атакам

Изначально задачей CDN являлось обеспечение требуемой производительности веб-сервера в периоды пиковой нагрузки. Эффекты Flash crowd и Slashdot — названия одного явления, когда сайт получает слишком много внимания пользователей и не может справиться с нагрузкой, — послужили важным толчком к появлению и развитию CDN. Безусловно, сегодня CDN обеспечивают значительное улучшение и других параметров, таких как задержка, оптимальное использование ресурсов, но распределение пиковой нагрузки по-прежнему остается одной из причин, по которой провайдеры контента используют услуги CDN.

«Пиковая нагрузка» может быть не чем иным как атакой отказа в обслуживании DoS. Эта угроза только увеличивается: мы наблюдаем постоянный рост числа и мощности атак, сила которых сегодня достигает сотен гигабит в секунду. И конечно, CDN играют существенную роль в защите от таких угроз. Благодаря облачной распределенной структуре крупная CDN может «поглотить» и переработать большие объемы трафика, тем самым обеспечивая доступность контента даже во время атак DDoS.

Неслучайно многие компании, предоставляющие услуги CDN, также предоставляют и услуги предотвращения атак DDoS. При этом CDN может предложить нечто большее, чем просто «поглощение» трафика. Современные «митигаторы атак DDoS» обеспечивают так называемый скраббинг (scrubbing) трафика — очистку полезного трафика от вредоносного. Отличить один от другого не так просто, поэтому здесь используются решения, основанные на подходах к защите от спама и вирусов, — анализ структуры трафика на предмет обнаружения известных шаблонов.

Защитный экран веб-приложений (Web Application Firewall, WAF)

Помимо защиты от атак DDoS многие CDN имеют в своем арсенале дополнительные услуги безопасности, такие как WAF. WAF — это защитный экран для приложений HTTP(S). Он применяет набор правил к обмену данными HTTP(S), который охватывает распространенные атаки, такие как межсайтовый скриптинг (XSS) и SQL-инъекция. WAF развернуты на граничных узлах, поэтому они могут останавливать вредоносный трафик ближе к источнику атаки, защищая исходный сервер.

Политика WAF — это свод правил, которые можно настраивать и устанавливать в разных местах CDN. Эти правила включают в себя определение списков разрешенных и заблокированных IP-адресов нежелательных источников запросов, контроль доступа на основе географического положения, правила, основанные на определенных параметрах HTTP-запроса, а также правила ограничения скорости. CDN может предлагать настраиваемые правила, предоставляемые клиентом, вместе с набором правил по умолчанию, например, обеспечивая защиту от 10 основных угроз OWASP[15].

[15] https://owasp.org/Top10

Интернет вещей

> Наиболее совершенными технологиями являются те, которые исчезают. Они все сильнее вплетаются в ткань ежедневной жизни до тех пор, пока не становятся неотличимы от нее.
>
> *Марк Вайзер, «Компьютер XXI века», 1991 г.*

В 1991 году в своем очерке «Компьютер XXI века» Марк Вайзер (Mark Weiser) предвосхитил появление Интернета вещей: «Компьютеры в выключателях, термостатах, музыкальных центрах и духовках уже помогают активировать мир. Эти и другие машины будут соединены между собой во всепроникающую сеть». Он продолжал:

«Присутствие сотни компьютеров в комнате может показаться пугающим на первый взгляд, так же в свое время вселяли страх сотни вольт, бегущих через провода в стенах. Но, как и провода в стенах, эти сотни компьютеров станут для нас незаметны. Люди будут просто использовать их бессознательно для выполнения повседневных задач».

Термин «Интернет вещей» был рожден позже, но Вайзер с удивительной точностью предсказал новый виток эволюции Интернета, когда многочисленные и незаметные устройства, объединенные в единое коммуникационное пространство, помогут перевести наши возможности по сбору, анализу и доступу к данным на качественно новый уровень, преобразуя эти данные в информацию и знание.

«Наиболее важным является то, что вездесущие компьютеры помогут преодолеть проблему информационной перегрузки. Объем информации, доступный нам во время прогулки по лесу, превосходит возможности любой компьютерной системы, и, тем не менее, мы находим прогулку среди деревьев расслабляющей, а работу с компьютером разочаровывающей. Машины, которые встраиваются в человеческую среду, не заставляя нас встраиваться в их собственную, сделают использование компьютера таким же освежающим, как и прогулка в лесу».

В этом разделе речь пойдет о вездесущих компьютерах, которые встраиваются в окружающие нас вещи — от осветительных приборов до одежды и аксессуаров, от кухонных аппаратов до автомобилей — и которые также интегрированы в информационную среду Интернета.

От Интернета сетей к Интернету вещей

История этого витка эволюции Интернета коротка. Говорят, что сам термин «Интернет вещей» впервые появился в 1999 году в презентации соучредителя центра Auto-ID при MIT Кевина Аштона (Kevin Ashton). Центр занимался разработкой технологии RFID и сенсоров, основанных на ней, а в презентации Кевин описал идею использования сенсоров, связанных между собой с помощью Интернета, в процессе производства и распределения продукции.

В то же время концепция использования сенсоров в индустриальных системах не нова. Например, в конце 1970-х некоторые электрические сети использовали коммерчески доступные счетчики, управляемые удаленно по телефонным линиям. В начале 1990-х существовали индустриальные системы, использовавшие достижения в беспроводных технологиях для обмена информацией между отдельными элементами — так называемое взаимодействие M2M (Machine to Machine).

Однако до недавнего времени сенсоры не были встроены в повседневные вещи и были ориентированы на индустриальные решения, а не на потребителя. Для ограниченного взаимодействия между собой они использовали закрытые решения и протоколы. Можно сказать, что они отличались от сегодняшнего феномена Интернета вещей, как арифмометр отличается от компьютера.

Но произошел качественный скачок. Как часто бывает в процессе эволюции, этому способствовали несколько факторов, набравших силу примерно в одно и то же время.

Повсеместная связность. Развитие сетевой инфраструктуры, а также доступность беспроводных решений с использованием мобильной и беспроводной связи делает почти все «подключаемым». Использование общего знаменателя — протокола IP — позволяет мгновенно интегрировать подключенное устройство в глобальную сеть Интернет. Удешевление связи, часто с нулевыми затратами на подключение дополнительного устройства (например, в домашней беспроводной сети), стимулирует проникновение связности в самые невообразимые материальные объекты.

Развитие компьютерных технологий. Закон Мура, предсказывающий экспоненциально возрастающую во времени производительность компьютеров, продолжает править компьютерной индустрией. Мощность процессоров все возрастает, а их размер, как и стоимость, все уменьшаются. Результат — появление миниатюрных устройств-сенсоров с компьютерным интеллектом, доступных потребителю.

Прогресс в области обработки данных. Появление новых алгоритмов анализа данных, а также облачных архитектур для их хранения и обработки при снижении стоимости позволяет существенным образом снизить ограничения на объем собираемой информации. Это, в свою очередь, открывает новые возможности для внедрения большего числа новых сенсоров в новых местах. А создаваемый сенсорами поток данных может быть эффективно обработан и проанализирован, порождая невообразимые объемы новой информации и знаний. Эта информация, в свою очередь, может быть использована для управления самими вещами и для влияния на поведение людей.

Несмотря на то, что сегодня термин «Интернет вещей», или IoT (Internet of Things), материализовался в коммерческих устройствах и системах и используется повсеместно, его общепринятого определения не существует.

Например, некоторые определяют IoT как подключение физических объектов к Интернету и связь их между собой с помощью миниатюрных встроенных сенсоров и проводных и беспроводных технологий для создания экосистемы всепроникающего компьютинга. Другие делают упор на встроенный интеллект в материальных объектах, позволяющий регистрировать изменение их состояния и соответствующим образом реагировать. Отсутствие единой трактовки во многом объясняется тем, что эта область нова и изменчива, к тому же эта тема затрагивает социальные аспекты и имеет как технический, так и философский характер.

Однако можно попробовать выделить несколько общих существенных элементов.

Сенсоры и контроллеры. Звук, движение, наблюдаемые и окружающие объекты, освещенность, температура — эти и другие параметры определяют состояние «вещи» и ее взаимодействие с окружающей средой. Если от «вещи» предполагается действие, она также содержит контроллер — регулятор или управляющее устройство. Так, например, дверь может распознать визитера по биометрическим параметрам и открыть замок, если они соответствуют хозяину жилища.

Отсутствующий пользовательский интерфейс. Большинство «вещей» получают информацию от сенсоров и управляющих серверов. Взаимодействие с пользователем часто происходит опосредованно, через управляющие серверы с интерфейсом порталов или путем использования приложений, с помощью которых пользователь может получить информацию о статусе объектов и задать определенные установки. Умная лампочка осветительной системы Hue компании Philips управляется не привычным диммером или выключателем, а с помощью приложения, установленного на вашем смартфоне или планшете.

Программируемый интеллект. По существу, «вещь», подключенная к Интернету, — это материализованное приложение. И, как в случае обычного приложения, ее функциональность может быть улучшена и расширена. Например, для осветительной системы Hue существует более сотни различных приложений, позволяющих управлять освещением в доме с учетом времени дня, года, музыки, текущей телевизионной программы и т.п. Подключив термостат Nest к Интернету, вы получите доступ к дополнительной функциональности, например, определению оптимального режима в зависимости от прогноза погоды. Проще говоря — границами возможностей является наше воображение.

Связность. Использование Интернета для обеспечения связности «вещей» позволяет им не только обмениваться информацией друг с другом или центральной системой. Интернет обеспечивает доступность «вещей» вне зависимости от вашего расположения. Вы можете управлять отопительной системой вашего дома из салона автомобиля, а климатической системой автомобиля — перед выходом из дома. Открытая коммуникационная инфраструктура Интернета позволяет таким системам обмениваться информацией с другими системами и информационными источниками. Например, система отопления может использовать данные прогноза погоды для выбора оптимального режима.

Автоматизация. Как уже было сказано, индустриальные сенсорные управляющие системы появились задолго до Интернета вещей. Уникальность сегодняшнего явления заключается в том, что оно принесло автоматизацию в массы, позволяя автоматизировать повседневные бытовые задачи. С другой стороны, благодаря открытой коммуникационной инфраструктуре могут быть также решены широкомасштабные задачи — от интеллектуальной транспортной системы до интеллектуального городского освещения.

Сотрудники Глобального института McKinsey выделили несколько областей применения Интернета вещей, благодаря которым можно ожидать дополнительный экономический рост. Наиболее важными, по их мнению, являются производственная и городская инфраструктура, гаджеты, торговля, транспортные системы и, конечно, домашнее хозяйство. Эти области представлены на рис. 83.

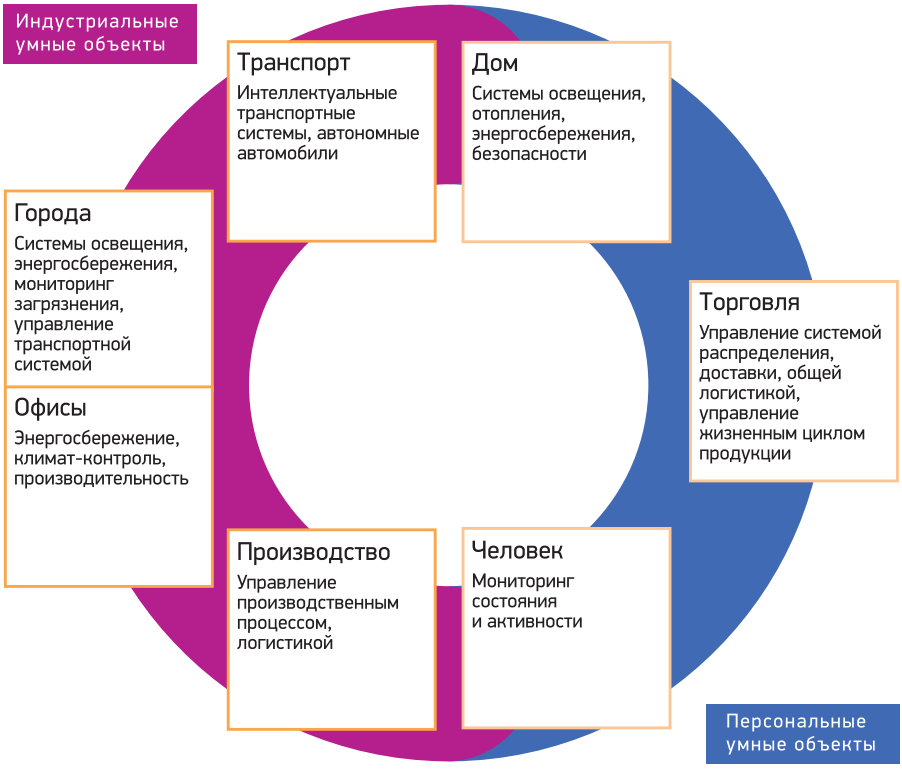

Рис. 83. Области применения IoT согласно данным Глобального института McKinsey.

Архитектурные модели

Мы только что рассмотрели IoT как концепцию и остановились на возможных определениях и отличительных чертах этого явления. Однако в конечном итоге каждый пример IoT — это система, включающая сенсоры и контроллеры, так называемые умные объекты, сети, промежуточные устройства, базы данных, управляющие серверы, порталы и приложения для взаимодействия и управления этими объектами.

Давайте рассмотрим несколько моделей, описывающих взаимодействие между различными компонентами, составляющими Интернет вещей.

Эталонная модель

Одна из моделей была разработана в МСЭ-Т и документирована в рекомендации Y.2060 «Обзор Интернета вещей».

Начнем с того, что рекомендация проводит границу между физическим и информационным мирами, предлагая следующие определения.

Применительно к Интернету вещей устройство означает элемент оборудования, который обладает обязательными возможностями связи и дополнительными возможностям измерения, срабатывания, а также ввода, хранения и обработки данных.

Применительно к IoT «вещи» — это предметы физического мира (физические вещи) или информационного мира (виртуальные вещи), которые могут быть идентифицированы и интегрированы в сети связи. К каждой «вещи» относится информация, которая может быть статической и динамической. Физические вещи существуют в физическом мире, их можно измерить, привести в действие и подключить. Примеры физических вещей — окружающая среда, промышленные роботы, товары и электрическое оборудование. Виртуальные «вещи» существуют в информационном мире, их можно хранить, обрабатывать, а также получать к ним доступ. Примеры виртуальных вещей — мультимедийный контент и прикладное программное обеспечение.

Физическая вещь интегрируется в информационное пространство с помощью устройств, которые взаимодействуют между собой. Устройства могут обмениваться данными с другими устройствами несколькими способами, показанными на рис. 84: с использованием сети связи через шлюз (сценарий a), с использованием сети связи без шлюза (сценарий b) или напрямую, то есть без использования сети связи (сценарий c). Кроме того, возможно сочетание сценариев a и c либо сценариев b и c. Например, устройства могут обмениваться данными с другими устройствами, используя прямую связь через локальную сеть — то есть сеть, обеспечивающую локальное соединение между устройствами и между устройствами и шлюзом, такую как специальная сеть (сценарий c), — и далее обмениваться данными с использованием сети связи через шлюз локальной сети (сценарий a).

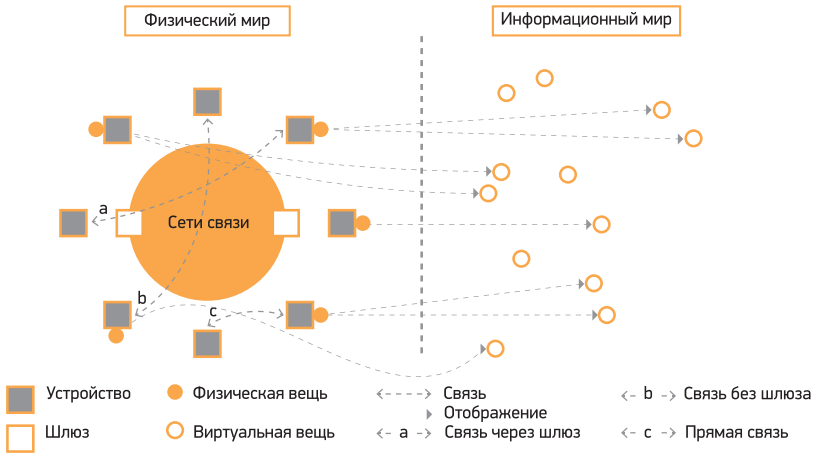

Рис. 84. Технический обзор IoT.
Источник: Рекомендация МСЭ-Т Y.2060 (06/2012)

Хотя на рис. 84 показаны только те взаимодействия, которые происходят в физическом мире (связь между устройствами), взаимодействия происходят и в информационном мире (обмен данными между виртуальными вещами), и между физическим миром и информационным миром (обмен данными между физическими и виртуальными вещами).

МСЭ-Т предлагает четырехуровневую эталонную модель, показанную на рис. 85.

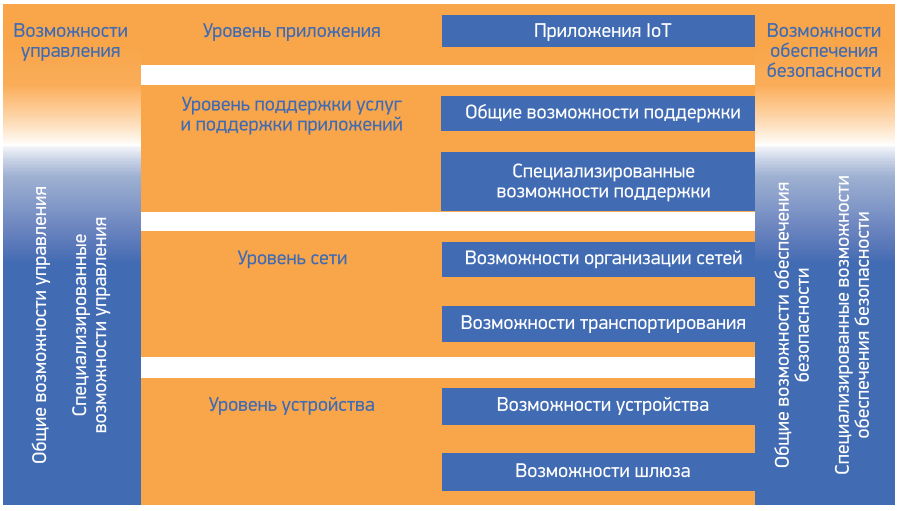

Рис. 85. Эталонная модель IoT МСЭ-Т.
Источник: Рекомендация МСЭ-Т Y.2060 (06/2012)

Самый верхний уровень содержит приложения IoT. Приложения взаимодействуют с уровнем поддержки услуг и поддержки приложений посредством программного интерфейса API.

Уровень поддержки услуг и поддержки приложений включает две группы возможностей:

1. Общие возможности поддержки — это типовые возможности, которые могут использоваться различными приложениями IoT, такие как обработка или хранение данных. Они могут быть активированы специализированными возможностями поддержки, например, для создания других специализированных возможностей поддержки.
2. Специализированные возможности поддержки — это конкретные возможности, которые предназначены для удовлетворения требований разнообразных приложений. В действительности они могут состоять из ряда групп четко определенных возможностей, для того чтобы предоставлять разные функции поддержки разным приложениям IoT.

Уровень сети включает два типа возможностей:

1. Возможности организации сетей: предоставляет соответствующие функции управления сетевыми соединениями, такие как функции управления доступом и ресурсом транспортирования, управление мобильностью или аутентификация, авторизация и учет (AAA).
2. Возможности транспортировки: предназначены для предоставления соединений для транспортировки информации в виде данных, относящихся к услугам и приложениям IoT, а также транспортировки информации контроля и управления, относящейся к IoT.

Наконец, на уровне устройства рассматриваются два вида возможностей в плане взаимодействия с сетью: возможности устройства и возможности шлюза. О возможностях устройства мы поговорим чуть подробнее ниже, в задачу же шлюза входит поддержка различных интерфейсов и трансляция протоколов, например, ZigBee и Bluetooth. При этом предусматриваются две ситуации:

первая ситуация возникает тогда, когда для связи на уровне устройства используются разные протоколы уровня устройства, например, протоколы технологий ZigBee и Bluetooth;

вторая ситуация возникает тогда, когда для связи, требующей и уровня устройства, и уровня сети, используются разные протоколы, например, протокол технологии ZigBee на уровне устройства и протокол технологии 4G на уровне сети.

В IoT минимальным требованием к устройствам является поддержка ими возможностей связи. В Рекомендации устройства подразделяются на несколько

категорий: устройства переноса данных, устройства сбора данных, сенсорные и исполнительные устройства, а также устройства широкого назначения. Они описываются следующим образом.

Устройство переноса данных: подключается к физической вещи и непрямым образом соединяет эту физическую вещь с сетями связи. Примером таких устройств являются активные теги RFID.

Устройство сбора данных: считывающее/записывающее устройство, имеющее возможность взаимодействия с физическими вещами. Взаимодействие может осуществляться непрямым образом с помощью устройств переноса данных или напрямую с помощью носителей данных, подключенных к физическим вещам.

Примером последних являются сканеры штрих- и QR-кодов.

Сенсорное и исполнительное устройство: может получать информацию об окружающей среде, например, о температуре или расположении в пространстве, и преобразовывать ее в цифровые электрические сигналы. Оно может также преобразовывать управляющие сигналы, поступающие от других компонентов IoT посредством сетей, в действия. Как правило, сенсорные и исполнительные устройства образуют закрытые сети, обмениваются друг с другом данными с помощью проводных или беспроводных технологий связи и используют шлюзы для подключения к Интернету.

Устройство общего назначения: обладает встроенными возможностями обработки и связи и может обмениваться данными с Интернетом с использованием проводных или беспроводных технологий. Устройства общего назначения включают оборудование и приборы, относящиеся к различным областям применения IoT, например, станки, бытовые электроприборы и смартфоны.

Примером последних являются сканеры штрих- и QR-кодов.

Сенсорное и исполнительное устройство: может получать информацию об окружающей среде, например, о температуре или расположении в пространстве, и преобразовывать ее в цифровые электрические сигналы. Оно может также преобразовывать управляющие сигналы, поступающие от других компонентов IoT посредством сетей, в действия. Как правило, сенсорные и исполнительные устройства образуют закрытые сети, обмениваются друг с другом данными с помощью проводных или беспроводных технологий связи и используют шлюзы для подключения к Интернету.

На рис. 86 показаны различные типы устройств и их взаимодействие в соответствии с моделью МСЭ-Т.

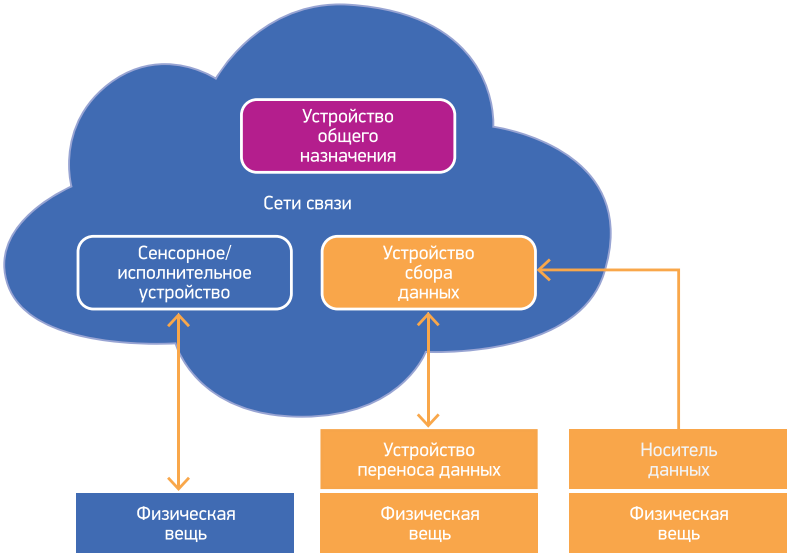

Рис. 86. Типы устройств и их взаимосвязь с физическими вещами.
Источник: Рекомендация МСЭ-Т Y.2060 (06/2012)

Коммуникационные модели IoT

Модели МКЭ-Т предлагают полезную классификацию компонентов экосистемы IoT, однако могут показаться немного абстрактными. Для более практического обсуждения возможных способов взаимодействия между устройствами и другими компонентами системы обратимся к документу IAB «Архитектурные соображения относительно сетей умных объектов», опубликованному в марте 2015 года в виде RFC 7452[16]. Этот документ выделяет несколько коммуникационных моделей, используемых в системах IoT, которые мы рассмотрим ниже.

Взаимодействие устройство–устройство

В рамках этой модели два или более устройств обмениваются между собой данными непосредственно, без участия каких-либо промежуточных элементов. В большинстве случаев связь между устройствами является беспроводной. Для обмена данными используются такие протоколы, как Bluetooth Smart, Z-wave или ZigBee. Простейший пример такой коммуникационной модели — умные лампа и выключатель, которые обмениваются данными о статусе и управляющими командами (рис. 87). Эта модель используется, когда сеть умных объектов является автономной и не требует глобальной связности.

[16] RFC 7452: Architectural Considerations in Smart Object Networking, URL: https://www.rfc-editor.org/rfc/rfc7452

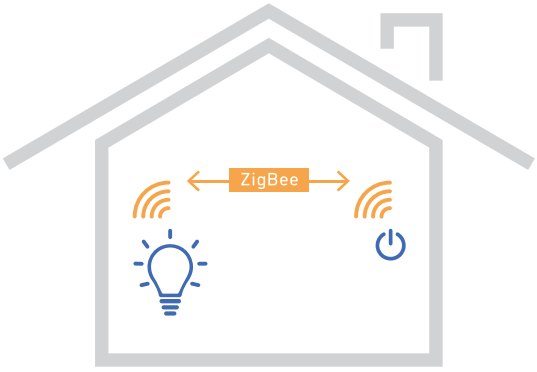

Рис. 87. Пример взаимодействия устройство–устройство.

Поскольку различные коммутационные протоколы несовместимы, устройства такой сети должны использовать один и тот же протокол. Например, семейство устройств с использованием протокола Z-Wave несовместимо с семейством ZigBee-устройств. Это, безусловно, ограничивает пользовательский выбор, хотя в рамках одного семейства, как правило, достигается хорошая совместимость.

Сети могут быть одноранговыми (P2P, peer-to-peer) либо иметь топологию «звезда». На первый взгляд, для решения таких задач, как автоматизация жилища, звезда является наиболее простой топологией с центральной базовой станцией и умными объектами, подключенными к ней. Так обычно реализуется домашняя беспроводная Wi-Fi-сеть. Однако различные помехи, например, трубы и перекрытия, зачастую не позволяют обеспечить связь со всеми объектами из одной центральной точки. Это еще более справедливо в географически распределенных объектах, таких как производство или система водо- или энергоснабжения в масштабах жилого блока или района.

Одноранговые сети, также называемые полносвязными (mesh network) в этом контексте, могут формировать произвольные структуры соединений, и их расширение ограничено только дистанцией между каждой парой узлов. В отсутствии прямой связности два устройства могут обмениваться данными между собой посредством третьего, связанного с первыми двумя. Эта технология позволяет создавать адаптивные беспроводные сети, способные к самоуправлению и самоорганизации.

Взаимодействие устройство–облако

В рамках этой модели устройство непосредственно взаимодействует с провайдером приложения, часто использующего облачные сервисы. Очевидно, что устройство должно поддерживать протокол IP и при этом его связность мало отличается от любого другого устройства — компьютера, смартфона и т.п., подключенного к Интернету. В качестве протоколов нижнего уровня устройства могут использовать IEEE 802.11 (Wi-Fi), что является типичным для домашней

беспроводной сети. Возможно и использование протокола ZigBee IP или Thread, если домашний беспроводной маршрутизатор или базовая станция поддерживают этот стандарт, а сеть поддерживает IPv6.

Пример применения этой модели — уже упоминавшийся обучающийся термостат Nest. Термостат передает данные на удаленный сервер для анализа энергопотребления. Удаленный сервер также является управляющим центром термостата, доступ к которому обеспечивается с помощью приложения, установленного, например, на смартфоне пользователя. Приложение Nest предоставляет пользователю полный доступ ко всем его продуктам Nest. Например, пользователь может изменить температуру или просмотреть потребление энергии у себя дома. Эта модель взаимодействия показана на рис. 88.

Использование провайдера приложений позволяет существенно расширить возможности самой системы в целом. Значительная часть интеллектуальной функциональности может быть реализована провайдером, а у устройства останутся лишь самые необходимые функции — сенсорные, управляющие и коммуникационные. Это также позволяет миниатюризировать устройство, снизить его стоимость и продлить срок службы батареи в случае автономного режима работы.

Например, термостат Nest может функционировать без связи с Интернетом, но его функциональность будет ненамного отличаться от традиционного, пусть и продвинутого термостата.

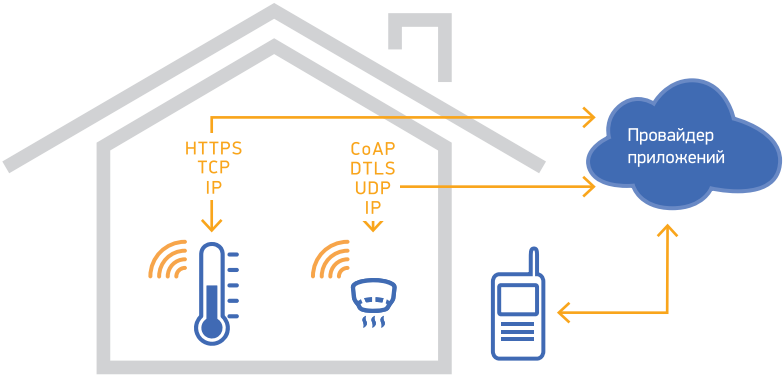

Рис. 88. Пример взаимодействия устройство–облако.

Многие успешные системы IoT представляют так называемые вертикали: когда устройства и облачные услуги провайдера приложений принадлежат одной и той же компании. И хотя порой для связности используются стандартные интернет-протоколы, на более высоких уровнях используются закрытые решения. Иногда провайдер приложений предоставляет API для доступа к облачным

услугам и опосредованного управления устройствами. Это позволяет расширить спектр разработчиков приложений для системы IoT и повысить инновационный потенциал.

Взаимодействие устройство–шлюз

В этом случае устройства взаимодействуют не с провайдером приложений, а с локальным шлюзом, обычно также выполняющим функции шлюза приложений (Application level gateway, ALG). Программное обеспечение шлюза приложений позволяет обмениваться данными с устройствами, управлять ими, обеспечивает безопасность и связь с провайдером приложений. Эта модель представлена на рис. 89.

Наиболее типичный пример реализации данной модели — носимые устройства, такие как фитнес-гаджеты. Сами устройства имеют весьма ограниченную функциональность и зачастую не поддерживают IP. В качестве шлюза обычно используется смартфон с установленным соответствующим приложением. Связь между устройством и смартфоном устанавливается, например, с помощью протокола Bluetooth. В свою очередь приложение смартфона имеет возможность синхронизировать данные с провайдером приложений и получить доступ к дополнительным данным, таким как отчеты и статистика физической активности.

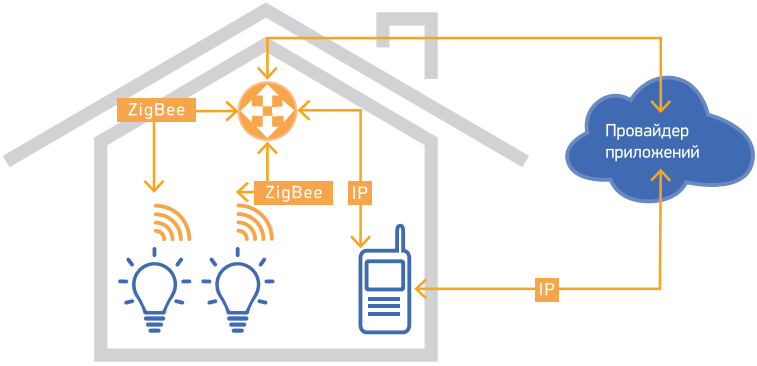

Рис. 89. Пример взаимодействия устройство–шлюз.

Другим типом шлюзов, часто встречающихся в системах автоматизации домашнего хозяйства, являются так называемые хабы, обеспечивающие связь со многими умными объектами в доме. Иногда хабы достаточно интеллектуальны для обеспечения всех функций системы и используют облачные услуги лишь для обеспечения удаленного доступа к системе. Примером такой схемы является система освещения Hue компании Philips. Хаб Hue Bridge обеспечивает управление всеми осветительными устройствами с помощью протокола ZigBee. Hue Bridge также подключается к домашней сети Wi-Fi и доступен через приложение на смартфоне пользователя для задания установок.

Другим примером является хаб SmartThings от Samsung. В этом случае он выполняет дополнительную функцию — обеспечивает обмен данными с устройствами, использующими различные протоколы, например, Z-Wave и ZigBee.

Сети

Одним из важнейших компонентов системы IoT является коммуникационная сеть. Это, как правило, беспроводная сеть, которая должна удовлетворять требованиям и ограничениям устройств, к ней подключенных. В частности:

- Энергосбережение. Многие устройства IoT функционируют в автономном режиме без внешнего источника питания. Срок службы батареи без подзарядки определяется годами, а иногда и десятилетиями.
- Незначительные компьютерные ресурсы, в частности, объем памяти и производительность процессора.

Эти ограничения подключенных устройств и сценарии использования IoT (например, для автоматизации дома и офиса, реализации системы освещения в «умных городах» или системы снабжения в розничной торговле) в свою очередь накладывают на сеть свои ограничения. Ими, в частности, могут являться:

- небольшая пропускная способность;
- возможный высокий процент потери пакетов;
- низкая энергия сигнала;
- асимметричные каналы;
- динамичная топология.

Обычно радиоэлемент (приемно-передающее устройство) является основным потребителем энергии. Также следует иметь в виду, что в энергоэффективных сетях (low power networks) передача данных столь же энергозатратна, как и функционирование в режиме прослушивания. Поскольку энергосбережение является одним из основных требований, накладываемых на устройства IoT, в идеале радиоэлемент должен включаться только в момент передачи данных.

Однако это требование накладывает существенное ограничение на тип сети, которая может быть создана. А именно — в полной мере подойдет только звездообразная сеть с центральным координирующим узлом и периферийными узлами-устройствами. В звездообразной сети радиоэлемент центрального узла включен все время. Это не проблема, поскольку данный узел обычно имеет внешний источник питания. Периферийные узлы, которые питаются от батарей, держат свои радиоэлементы выключенными и включают только в момент необходимости передачи данных. Очевидно, что единственным узлом, которому данные могут быть переданы, является центральный узел.

Звездообразная топология является наиболее простым типом сети, но она имеет существенный недостаток: ограниченный радиус действия. Действительно, для

полноценного функционирования сети все объекты должны находиться в зоне «видимости» центрального узла. Во многих сценариях реализации IoT это не всегда осуществимо.

Для возможности динамического расширения диапазона сети узлы должны выполнять функцию радиотрансляции — принимать и передавать данные между собой, не полагаясь на центральный узел. В этом случае используется так называемая полносвязная сеть (mesh network). За счет ретрансляции протяженность сети может быть значительно увеличена путем добавления дополнительных узлов. Также такая топология предполагает избыточные пути через сеть, обеспечивая повышенную надежность. При выходе какого-либо узла из строя трафик может быть передан в обход.

В такой топологии все узлы могут разговаривать друг с другом, формируя надежную многоскачковую сеть. Но, учитывая требования энергосбережения, радиоэлементы узлов должны управляться таким образом, чтобы они выключались, когда нет трафика, но включались, когда соседи хотят общаться.

Для этого обычно используются два основных подхода — энергосберегающее прослушивание (Low Power Listening, LPL) и синхронизированный по времени ячеистый протокол (Time Synchronized Mesh Protocol, TSMP).

Энергосберегающее прослушивание LPL
При использовании этого подхода узлы работают в импульсном режиме, периодически включая радиоэлемент на непродолжительное время в попытке засечь возможные сигналы о желании передать данные от других соседних узлов. Скважность режима зависит от конкретного типа системы и трафика. Типичным является следующий режим: полсекунды неработы, чередующиеся с несколькими сотнями миллисекунд рабочего состояния.

Устройство, желающее передать данные, сначала посылает стробоскопический сигнал, достаточно продолжительный, чтобы перекрыть период импульсного включения радиоэлемента других узлов. При получении строба при очередном включении радиоэлемента узел оставляет его включенным в ожидании передаваемых данных.

Хотя метод очень прост и не требует явной синхронизации всех узлов сети, он имеет ряд недостатков. Во-первых, стробы пробуждают все узлы, которых они достигают. Во-вторых, чем длительнее период между импульсами включения, тем больше сохраняется энергии, но в то же время тем больше требуется времени на передачу сигнала, поскольку отправителю необходимо сначала передать достаточно долгий стробоскопический сигнал.

Чтобы частично исправить эти недостатки, стробоскопический сигнал содержит идентификатор адресата, таким образом пробуждая только требуемый узел. Также, предполагая постоянство периода импульсов, отправитель может точнее

определить время включения радиоэлемента и, соответственно, непосредственно перед этим послать стробоскопический сигнал. Работа протокола схематически показана на рис. 90(a).

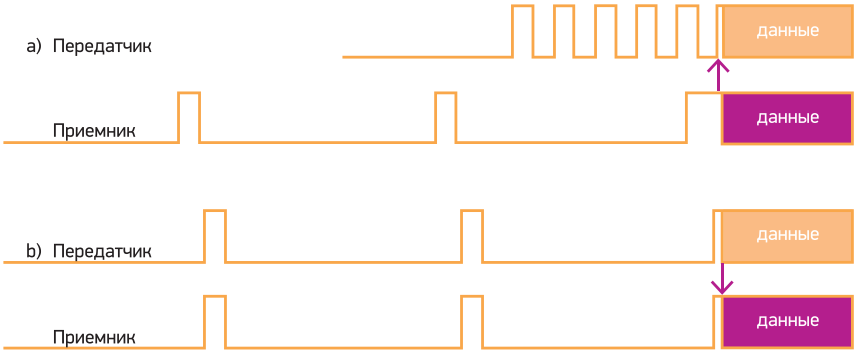

Рис. 90. Работа передающего и принимающего устройства в случае использования LPL (a) и TSMP (b).

Протокол TSMP

Недостатки асинхронных протоколов помогает исправить явная синхронизация всех устройств сети. В этом случае достигается максимальное энергосбережение, так как устройства включают приемники и передатчики на очень короткий промежуток времени. Для синхронизации устройств используется протокол временной синхронизации, например, TSMP (Time Synchronized Mesh Protocol, протокол синхронизации полносвязной сети).

В соответствии с этим протоколом все устройства синхронизированы с точностью до 50 мкс. В начале каждого временного слота продолжительностью 10 мс все устройства кратковременно включают радиоэлементы. Если устройству необходимо передать данные в текущем временном слоте, оно должно начать передачу в течение 100 мкс с его начала. Таким образом, в общем случае принимающие устройства должны держать свои радиоэлементы включенными не дольше 100 мкс на каждые 10 мс. Работа протокола схематически показана на рис. 90(b).

Протокол TSMP требует централизованного управления сетью. Он был разработан для индустриальных областей внедрения IoT и не используется в сетях автоматизации дома и офиса.

Стандарты

В зависимости от характера применения систем IoT для связи объектов используются различные сетевые технологии и протоколы. Отчасти это связано с развитием проприетарных решений, созданием так называемых вертикалей,

когда ноу-хау является частью конкурентоспособности системы. Кроме того, системы IoT значительно различаются по своим характеристикам и требованиям к автономности функционирования, географической протяженности, объемам передаваемых данных и т.п. Например, радиус действия систем автоматизации дома и офиса, как правило, не превышает нескольких десятков метров, в то время как системы, внедряемые в «умных городах» (например, системы освещения, контроля транспорта и энергосбережения), требуют дальности передачи, измеряемой десятками километров.

Мы уже кратко остановились на некоторых стандартах, используемых в IoT. Основные усилия здесь направлены на разработку беспроводных технологий физического и канального уровней, которые бы удовлетворяли требованиям и ограничениям, связанным с IoT. Все беспроводные протоколы, за исключением мобильной связи, используют нелицензируемый спектр. На рис. 91 представлены стеки нескольких наиболее популярных протоколов и технологий. Поговорим о них подробнее.

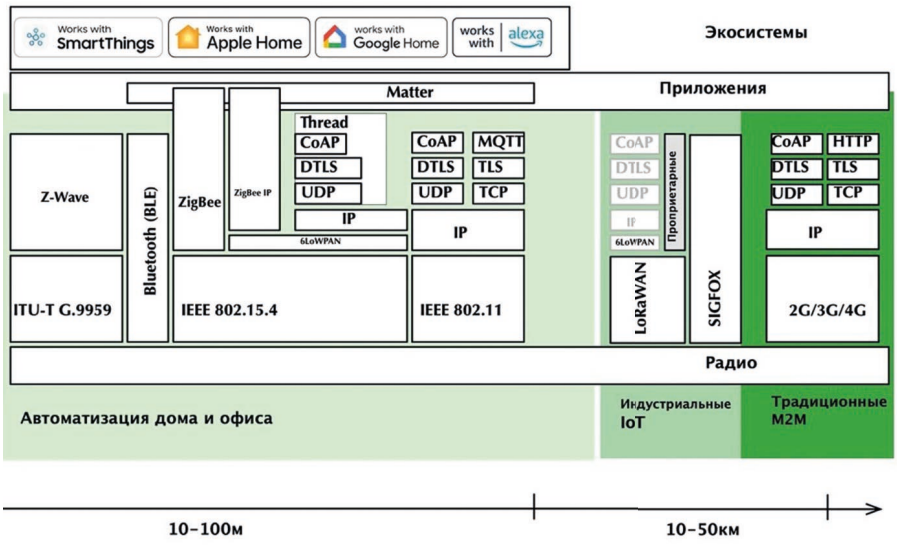

Рис. 91. Стеки популярных протоколов IoT.

Bluetooth

Bluetooth, пожалуй, является самым старым и знакомым протоколом для связи между различными устройствами на коротких расстояниях. Изначально разработанный компанией Ericsson в 1994 году в качестве беспроводной альтернативы серийному кабелю RS-232, этот протокол используется в широком спектре компьютерных устройств. Протокол был стандартизован IEEE и в настоящее время обслуживается организацией Bluetooth Special Interest Group.

Этот протокол доминирует в системах управления и обмена информацией с различными гаджетами — от фитнес-сенсоров до традиционных наушников и спикеров. Обмен данными обычно происходит между гаджетом и смартфоном, выполняющим роль шлюза. Новая версия протокола — Bluetooth Low-Energy (BLE), или Bluetooth Smart, — является важным протоколом для приложений IoT. Обеспечивая диапазон действия, аналогичный традиционному Bluetooth, BLE позволяет значительно сократить потребление электроэнергии.

При поддержке профиля Internet Protocol Support и адаптационного уровня 6LoWPAN, о котором мы поговорим чуть позже, устройства способны непосредственно использовать протокол IPv6. Благодаря этому системы IoT могут быть напрямую подключены к Интернету для управления умными объектами.

IEEE 802.11

Беспроводные сети на основе протоколов семейства IEEE 802.11 составляют основу Wi-Fi-хот-спотов, офисных и домашних сетей. Этот стандарт поддерживается практически всеми переносными компьютерными устройствами — от смартфона до ноутбука и персонального компьютера. Скорости в 54 Мбит/с являются вполне обычными (IEEE 802.11a/g), но все больше сетей поддерживают более высокую пропускную способность в сотни Мбит/с (IEEE 802.11n/ac).

Энергосбережение не является отличительной чертой этих протоколов, поэтому они могут быть использованы в системах, где устройства подключены к источнику питания. Однако версия протокола IEEE 802.11ah является энергосберегающей и была специально разработана для систем IoT.

IEEE 802.15.4

Стандарт IEEE 802.15.4 разработан и поддерживается рабочей группой IEEE 802.15. Он определяет физический уровень и уровень управления доступом к среде (Medium Access Control, MAC) для энергоэффективных беспроводных персональных сетей небольшой дальности действия. Максимальная скорость передачи, определенная стандартом, — 250 кбит/с, а выходная мощность равна 1 мВт. Номинальный радиус действия — несколько десятков метров.

Этот стандарт является «строительным блоком» для нескольких других протоколов IoT, таких как ZigBee, ZigBeeIP и Thread, определяющих верхние слои стека вплоть до уровня приложений.

IEEE 802.15.4 может использоваться совместно со стандартом 6LoWPAN, который определяет адаптационный уровень IP. В этом случае в качестве верхних уровней стека могут применяться стандарты IETF — UDP, DTLS, CoAP, а устройства могут быть напрямую подключены к Интернету с глобальной связностью.

6LoWPAN

Стандарт 6LoWPAN разработан и поддерживается IETF. Он определяется спецификациями RFC 4944[17], RFC 6282[18] и RFC 6775[19]. С помощью этого стандарта пакеты IPv6 могут передаваться по сетям IEEE 802.15.4. Дело в том, что стандартные пакеты IPv6 слишком велики для энергоэффективных беспроводных сетей IoT, таких как IEEE 802.15.4.

Стандарт IEEE 802.15.4 предоставляет всего 81 байт для протоколов верхнего уровня, что значительно меньше максимального размера пакета (Maximum Transmission Unit, MTU) IPv6, не требующего фрагментации, — 1280 байт. Поскольку в стандарте IPv6 промежуточные устройства не осуществляют фрагментацию, эту функцию должен выполнять шлюз между сетью IEEE 802.15.4 и Интернетом. Стандарт определяет адаптационный уровень между IEEE 802.15.4. и IP, обеспечивающий необходимую фрагментацию и сбор пакетов. Но это еще не все. Только заголовок IPv6 занимает почти половину доступного места (40 байт), что, с учетом заголовков фрагментации и накладных расходов протоколов более высокого уровня, практически не оставляет места для полезных данных. Решением является компрессия заголовка, определенная в RFC 4944 и RFC 6282. Предложенные алгоритмы позволяют сжать заголовки IPv6 и UDP до 12 байт.

ZigBee

ZigBee — один из популярных протоколов, применяющихся в системах домашней и офисной автоматизации, например, в системах освещения. Система Hue компании Philips использует этот протокол для обмена данными с осветительными приборами и шлюзом Hue Bridge.

ZigBee использует протокол IEEE 802.15.4 для нижних уровней стека, но определяет собственное шифрование и маршрутизацию данных. Нужно заметить, что протокол ZigBee также обслуживает уровень приложений и, соответственно, имеет несколько профилей для различных сценариев применения: домашняя автоматизация, энергетика, здравоохранение и т.д. Новый стандарт, ZigBee 3.0, унифицирует обмен данными между этими кластерами приложений, облегчая интеграцию различных систем.

Для некоммерческого использования спецификации ZigBee доступны бесплатно. Для производителей требуется вступление в ассоциацию ZigBee Alliance.

---

17 RFC 4944: Transmission of IPv6 Packets over IEEE 802.15.4 Networks, URL: https://www.rfc-editor.org/rfc/rfc4944

18 RFC 6282: Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks, URL: https://www.rfc-editor.org/rfc/rfc6282

19 RFC 6775: Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), URL: https://www.rfc-editor.org/rfc/rfc6775

Как и у многих протоколов локальных энергоэффективных сетей, радиус действия ZigBee не превышает нескольких десятков метров, но может быть расширен за счет промежуточных устройств. Скорость передачи данных также невысока — 250 кбит/с.

ZigBee IP

ZigBee IP является версией протокола ZigBee, поддерживающей протокол IPv6 на сетевом уровне. Для этого ZigBee использует адаптационный уровень 6LoWPAN. Применение ZigBee IP позволяет осуществить сквозную адресацию устройств IoT-системы, обеспечив их непосредственную связность с Интернетом.

Z-Wave

Так же, как и ZigBee, Z-Wave является одним из наиболее популярных технологий систем IoT для автоматизации дома и офиса. Основу технологии составляет радиоэлемент и протокол компании Sigma Designs.

В качестве протокола нижних уровней Z-Wave использует ITU-T G.9959, реализуя верхние уровни вплоть до приложений стеком протоколов от Sigma Designs. Стандарт Z-Wave поддерживается ассоциацией Z-Wave Alliance. Z-Wave использует частоту 900 МГц и работает в топологии полносвязной сети.

Для разработчиков и производителей чипов Sigma Designs лицензирует дизайн, программное обеспечение стека и программный интерфейс. Спецификации Z-Wave доступны без роялти, на основе модели RAND (Reasonable and Non-discriminatory, https://ru.wikipedia.org/wiki/Reasonable_and_Non-Discriminatory).

Thread

Thread был впервые анонсирован в 2014 году коалицией компаний, включающих Nest Labs от Google, Samsung Electronics и ARM. Разумеется, первым устройством, поддерживающим этот стандарт, был термостат Nest, но число совместимых систем неуклонно растет.

Thread, пожалуй, самый открытый стандарт из всех перечисленных. Как и ZigBee, он использует стандарт IEEE 802.15.4 для нижних уровней стека, а также адаптационный уровень 6LoWPAN для поддержки IPv6. Другими словами, все устройства Thread используют IPv6 в качестве сетевого протокола — а значит, эти устройства органично интегрированы в домашнюю Wi-Fi-сеть и Интернет.

Thread позволяет создать одноранговую сеть взаимодействующих друг с другом устройств. Это означает, что все устройства Thread обмениваются данными друг с другом (вместо связи только с базовой точкой доступа, как в Wi-Fi-сетях), что обеспечивает повышенную надежность и более широкий охват. Хотя бы одно из устройств должно выполнять функции «граничного маршрутизатора», обеспечивая связь этой одноранговой сети с домашней сетью.

Matter

Одной из проблем современных IoT является совместимость на уровне приложений. Обратите внимание на знаки «работает с...» на многих устройствах умного дома сегодня. «Работает с Apple Home», «работает с SmartThings», «работает с Alexa», «работает с Google Home» - каждый такой значок указывает, какую экосистему умного дома поддерживает это устройство. В рамках одной экосистемы устройства работают безупречно, но попробуйте подключить «чужестранца», и вы сразу столкнетесь с проблемой совместимости. Скорее всего, интегрировать такое устройство не получится, и вам придется управлять им специально для него созданным приложением.

С пользовательской точки зрения такая ситуация нежелательна, да и для производителей устройств и даже создателей экосистем она не оптимальна. Поэтому несколько лет назад Apple, Google, Amazon, Samsung и другие объединились через Альянс стандартов связности (Connectivity Standards Alliance, CSA; который ранее назывался Альянсом ZigBee), чтобы попытаться создать единый стандарт, который позволил бы одному устройству работать через множественные экосистемы. Эта работа завершилась в 2022 году созданием стандарта под названием Matter.

Matter работает поверх существующих сетевых и транспортных протоколов, поддерживая Thread, Wi-Fi, Ethernet и BLE. С другой стороны, Matter определяет стандартный набор команд для управления и интеграции устройств в экосистемы IoT.

Таблица 8. Сравнительные характеристики популярных технологий беспроводной связи в системах IoT

	Bluetooth BLE	ZigBee	Z-Wave	Thread	IEEE 802.11
Дальность	50–150 м	10 м	30 м	30 м	50 м / 1 км(802.11ah)
Максимальное количество устройств сети	8 (пиконет)	65 000	232	300	2007
Скорость передачи	1 Мб/с	40–250 кб/с	9.6–100Кб/с	250 Кб/с	50–200 Мбит/с 100 кбит/с (802.11ah)
Частота	2,4 ГГц	915 МГц / 2,4 ГГц	900 МГц	2,4 ГГц	900 МГц (802.11ah) / 2,4 ГГц /5 ГГц
Тип сети	звездообразная	полносвязная, древовидная, звездообразная	полносвязная	полносвязная	звездообразная

Sigfox

Французская компания Sigfox была основана в 2009 году для построения инфраструктуры для различных распределенных приложений IoT, таких как системы освещения, индустриального контроля и т.п. Разработанная технология и инфраструктура также получила название Sigfox. Однако, истратив 350 миллионов евро инвестиций, Sigfox зарегистрировала всего около 20 миллионов подключений, и в январе 2022 года вынуждена была подать заявление о неплатежеспособности, не перенеся экономических последствий COVID-19. Спустя

несколько месяцев она была приобретена сингапурской компанией UnaBiz, которая с тех пор является владельцем технологии Sigfox oG. Эта технология используется более чем 70 oG-операторами по всему миру.

Технология Sigfox oG основана на сетевом протоколе с низким энергопотреблением (LPWA) и определяет полный пакет — от радиопротокола, программного обеспечения коммуникационных элементов устройств до коммуникационной инфраструктуры и облачного хранилища данных и доступа к данным систем IoT. Единственным более или менее открытым элементом системы является лицензирование радиотехнологии различным производителям. Например, STMicroelectronics, Atmel и Texas Instruments производят радиоэлементы для сети Sigfox oG.

Sigfox oG использует узкий диапазон радиоспектра, что обеспечивает лучшую защиту от шума даже при маломощной передаче. Оконечные устройства недорогие, особенно по сравнению с весьма дорогостоящими базовыми станциями, обслуживающими сеть.

Наибольшее покрытие Sigfox oG обеспечивает в странах Западной Европы, см. рис. 92, хотя сегменты сети разворачиваются глобально, включая США, Новую Зе-

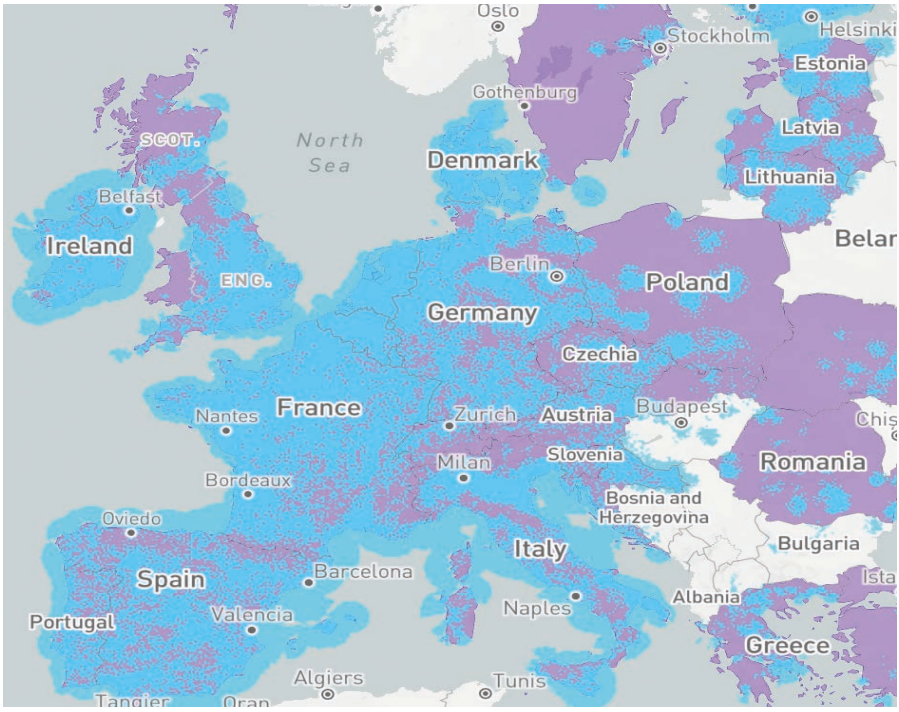

Рис. 92. Зона покрытия сетей Sigfox. Голубым цветом показано текущее покрытие, фиолетовым — области, где в настоящее время происходит развертывание технологии.
Источник: https://www.sigfox.com/coverage

ландию, Австралию, Бразилию и Оман. UniBiz работает с региональными сетевыми операторами, которые, собственно, и разворачивают эту технологию, выплачивая UniBiz роялти за ее использование. Другими словами, они выступают как реселлеры сетевых услуг и услуг приложений Sigfox 0G.

LoRaWAN

LoRaWAN отчасти напоминает Sigfox, хотя есть и существенные отличия, как в плане технологии — радио LoRa использует широкий спектр (обычно более 125 кГц), — так и в отношении бизнес-модели. LoRa является технологией радиомодуляции для сетей большой дальности с низким энергопотреблением и низкой скоростью передачи данных. Основные особенности данной технологии — большое количество устройств, которые могут работать в рамках одной локальной сети, и относительно большая дальность, которая может быть покрыта одним маршрутизатором LoRaWAN. Один шлюз способен координировать около 20 000 узлов в диапазоне 10–30 км.

LoRaWAN является более открытой системой. Спецификации LoRaWAN свободно доступны — любой производитель сетевого оборудования может начать выпускать модули или шлюзы для этой системы. Более того, сетевой оператор может создать собственную инфраструктуру, включая систему сбора, анализа и доступа к данным. Единственным ограничением является то, что радиоэлемент LoRa основан на чипе компании Simtech.

Разработкой и сопровождением стандартов, а также сертификацией оборудования занимается отраслевая ассоциация LoRa Alliance с достаточно широким спектром организаций-членов[20].

Стандарты сотовой связи

Технологии передачи данных сотовой связи 2G/3G/4G/5G могут использоваться для приложений IoT, требующих обмена информацией на больших расстояниях с использованием существующей инфраструктуры. Эти сети способны передавать большие объемы данных, особенно 5G, однако потребляемая мощность является недопустимо высокой для многих применений. Впрочем, для систем с устройствами, осуществляющими нечастый кратковременный обмен данными, эти технологии вполне приемлемы.

Вопросы безопасности и защиты персональных данных

Вместе с ростом числа и типов устройств, подключенных к Интернету, растут и риски, связанные с безопасностью и защитой частной жизни. Это особенно справедливо для IoT: окружающие нас вещи могут быть использованы не по назначению и в преступных целях, их функциональность может быть изменена вплоть до отказа работы. В то время как умные объекты все теснее вплетаются в нашу жизнь, делая нас все более зависимыми от них, вопросы обеспечения защищенности систем IoT, персональных данных становятся как никогда актуальными и составляют важный элемент нашей собственной безопасности и защищенности частной жизни.

[20] https://lora-alliance.org/member-directory

Несколько факторов усугубляют ситуацию:

- Индустрия умных объектов достаточно молодая. Многие компании являются либо стартапами, либо берут свои истоки в отраслях, отдаленных от компьютеров и Интернета. Это зачастую приводит к недостаточному знанию и опыту в вопросах компьютерной безопасности. Например, использование открытых каналов для обмена данными между устройствами является одним из встречающихся уязвимых мест таких систем.
- Основной упор делается на функциональность устройств и системы в целом. Учитывая желание минимизировать стоимость устройств, это зачастую приводит к недостаточному вниманию к вопросам безопасности.
- Масштаб внедряемых систем IoT существенен. Домашняя, офисная или индустриальная сеть теперь должна обслуживать на порядки больше устройств. Более того, однотипность этих устройств значительно усиливает эффект обнаружения уязвимости в одном из них.
- Некоторые устройства не имеют возможности автоматического обновления программного обеспечения с целью закрытия уязвимых мест. Учитывая продолжительный срок жизни многих устройств, отсутствие такой функциональности представляет существенный риск. Возьмем, например, инцидент с маркой Джип[21], когда была открыта уязвимость, позволявшая атакующему удаленно получить контроль над важными функциями автомобиля, включая тормоза. Компании Fiat Chrysler пришлось отозвать 1,4 миллиона машин для обновления программного обеспечения. Учитывая неудобства, которые эта операция доставляет многим владельцам, можно предположить, что в значительной части автомобилей уязвимость осталась незакрытой.
- Все больше окружающих нас вещей используют Интернет для расширения своей функциональности. Становится все сложнее приобрести «вещь», которая бы не подключалась к Интернету.
  Все более важные жизненные аспекты зависят от правильного функционирования таких систем. Например, «глюк» с термостатом Nest[22] чуть не заморозил владельцев этого устройства.
- Многие сенсоры производят сбор весьма конфиденциальных данных, предоставляющих информацию о наших привычках, поведении, нахождении, могут прослушивать наши разговоры и делать видеозаписи. Например, устройства SmartTV компании Samsung могут управляться голосовыми командами. Проблема заключается в том, что для этого телевизор пересылает услышанную речь в Samsung для анализа на предмет возможных команд[23]. Разумеется, это зачастую не только команды, но и просто подслушанный разговор. Насколько хорошо защищены эти данные, и кто имеет к ним доступ?

---

21 https://www.wired.com/2015/07/jeep-hack-chrysler-recalls-1-4m-vehicles-bug-fix

22 https://www.nytimes.com/2016/01/14/fashion/nest-thermostat-glitch-battery-dies-software-freeze.html?_r=0

23 https://www.cnet.com/news/privacy/samsungs-warning-our-smart-tvs-record-your-living-room-chatter

Причиной многих из этих проблем являются не IoT как таковой, а то, что цифровой мир и Интернет все плотнее интегрируются с физическим миром. Все больше персональных и конфиденциальных данных хранятся в «облаках», а мы все более зависимы от умных полезных устройств, приложений, Интернета. IoT, безусловно, делает некоторые из этих проблем значительнее.

Защищенность системы — не бинарное состояние: степени безопасности представляют собой широкий спектр. Защищенность системы также зависит и от характера угроз. Все эти факторы меняются во времени. Будем надеяться, что по мере того как отрасль становится более зрелой, безопасность IoT будет обеспечиваться на адекватном уровне.

Заключение

Взгляд в будущее открывает неполную и порой не совсем верную картину. Несмотря на то, что технология является мотором изменений в Интернете, успех того или иного нововведения определяется социально-экономическими факторами.

Как и наш физический мир, развитие Интернета подчиняется законам естественного отбора. Разрабатываются новые технологии и основанные на них решения — некоторые тихо умирают, некоторые распространяются с эпидемической скоростью. Благодаря глобальному покрытию и миллиардам участников в Интернете постоянно происходят мутации — новые приложения, протоколы, улучшенные версии существующих, — которые мгновенно становятся доступными в глобальном масштабе. Что-то приживается и пускает корни, что-то — нет.

Спроектировать Интернет невозможно. Его развитие определяется взаимодействием различных тенденций и интересов, а не согласованным планом или архитектурной моделью или даже технологией.

И в то же время этот процесс не является полностью случайным. Мы можем увидеть очертания будущего в технологиях и приложениях настоящего. Не думаю, что мы сильно ошибемся, если предположим, что тенденция увеличения пропускной способности и качества доступа к информации и, что более важно, к знанию будет продолжаться, превращая Интернет в гигантский суперкомпьютер. «Сеть — это компьютер» — слоган уже несуществующей компьютерной компании Sun Microsystems сегодня верен как никогда. Спроектировать Интернет невозможно. Но каждый из нас вносит вклад в его будущее.

Передовые операционные практики и рекомендации

> В настоящее время у IETF нет механизма или средств для публикации соответствующей технической информации, одобренной IETF. Этот документ создает новую подсерию RFC под названием Best Current Practices (BCP).
>
> *RFC 1818[1], август 1995 г.*

Разработка новых протоколов, решений и технологий является необходимым условием полноценного развития Интернета. Однако их ценность и эффект появляются только в процессе внедрения и использования. И не просто использования, а использования их значительной частью операторов связи и услуг.

Существенным фактором, усложняющим процесс внедрения новых протоколов и технологий, является децентрализованный характер Интернета. Ведь для успешного внедрения необходимо, чтобы все участники, обеспечивающие передачу данных, добровольно согласились применить нововведение. Особенно остро эта проблема стоит перед протоколами и технологиями инфраструктуры в области адресации, системы имен DNS, маршрутизации. Проблема отягощается не только числом участников, которые должны договориться, но и отсутствием «бизнес-кейса» или его слабостью: преимущества от внедрения технологии на начальном этапе незначительной группой участников не соответствуют уровню затрат и рискам, связанным с внедрением. Например, какая польза от протокола IPv6, если по этому протоколу доступно только незначительное число ресурсов,

[1] RFC 1818: Best Current Practices, URL: https://www.rfc-editor.org/rfc/rfc1818

а при этом оператору, по сути, необходимо параллельно управлять двумя сетями (IPv4 и IPv6)?

В этом плане значительную ценность представляют так называемые передовые практики. Они позволяют существенно снизить затраты и избежать ошибок при внедрении и эксплуатации новых технологий.

DNS

В этом разделе мы кратко остановимся на практиках и рекомендациях по внедрению и эксплуатации системы DNS.

Руководство по развертыванию защищенной системы доменных имен (DNS), специальная публикация NIST 800-81-2[2]

Это руководство, разработанное Национальным институтом стандартов и технологий США, содержит рекомендации по защите DNS на предприятии. В документе представлены подробные рекомендации по поддержанию целостности данных и аутентификации источника, которые необходимы для обеспечения подлинности информации о доменных именах и поддержания целостности информации о доменных именах при передаче. Также в документе представлены рекомендации по настройке DNS для предотвращения атак типа «отказ в обслуживании» (Denial of Service, DoS), использующих уязвимости в различных компонентах DNS. Это наиболее часто используемый тип атак, направленный на нарушение доступа к ресурсам, доменные имена которых обрабатываются атакованными компонентами DNS.

Рекомендации по противодействию DDoS-атакам с использованием инфраструктуры DNS, SSAC065[3]

Эти рекомендации, разработанные консультативным комитетом ICANN по безопасности и стабильности (Security and Stability Advisory Committee, SSAC), предлагают комплексный подход к противодействию распределенным рефлекторным атакам отказа в обслуживании с усилением (reflection-amplification DDoS attacks). Поскольку создание таких атак использует уязвимые места многих систем, обслуживаемых различными операторами, эффективное решение требует коллективных действий. Так, например, для успешного запуска атаки используются сети, в которых отсутствует защита от спуфинга адресов источника, DNS-резолверы с открытым доступом, а также авторитетные DNS-серверы. Документ предлагает список конкретных действий, адресованный операторам этих компонентов общей системы.

---

[2] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-81-2.pdf

[3] https://itp.cdn.icann.org/en/files/security-and-stability-advisory-committee-ssac-reports/sac-065-en.pdf

Предотвращение использования рекурсивных резолверов для рефлекторных атак, RFC 5358[4]

Рекурсивные резолверы являются привлекательным объектом для запуска рефлекторных DDoS-атак с усилением. Действительно, резолверы используют протокол DNS, основанный на UDP, не требующем создания канала, а размеры ответов могут в десятки раз превышать размеры запросов. Документ содержит рекомендации для предотвращения такого использования. Общая рекомендация операторам серверов имен — использовать средства, поддерживаемые выбранной реализацией, для предоставления услуги рекурсивного поиска имен только предполагаемым клиентам. Другими словами, серверы имен не должны предлагать рекурсивное обслуживание внешних сетей. Для серверов имен, выполняющих роль авторитетных серверов, функции рекурсивного резолвера должны быть отключены.

Рекомендации для операторов DNS со службой конфиденциальности, RFC 8932[5]

Этот документ предлагает рекомендации для операторов DNS-резолверов, предоставляющих услуги конфиденциальности. Под услугами конфиденциальности подразумевается шифрование трафика между клиентом и резолвером с использованием протоколов DoT [RFC7858], «DNS over DTLS» [RFC8094] и DoH. Документ содержит рекомендации по хранению и обработке данных для операторов служб конфиденциальности DNS. Также в документе рассматривается ситуация, когда оператор такого резолвера, особенно в случаях, когда используется резолвер, внешний по отношению к сети пользователя, имеет доступ ко всей информации, которую защищают протоколы шифрования. Усугубляется это тем фактом, что в ряде случаев пользователь продолжает использовать этот же резолвер при перемещении в другую сеть, тем самым предоставляя возможность трассировки.

Для увеличения прозрачности и ответственности операторов таких услуг документ предлагает шаблон «заявления о конфиденциальности рекурсивного оператора (Recursive operator Privacy Statement, RPS)». RPS — это документ, который должен опубликовать оператор. В документе описываются методы работы оператора и обязательства в отношении конфиденциальности, что дает клиентам возможность оценить как измеримые, так и заявленные свойства конфиденциальности конкретной службы конфиденциальности DNS.

---

4 RFC 5358: Preventing Use of Recursive Nameservers in Reflector Attacks, URL: https://www.rfc-editor.org/rfc/rfc5358

5 RFC 8932: Recommendations for DNS Privacy Service Operators, URL: https://www.rfc-editor.org/rfc/rfc8932

Сетевая инфраструктура

В этом разделе мы кратко остановимся на практиках и рекомендациях по эксплуатации сетевой инфраструктуры и ее адресации. Особое внимание уделено практикам по развертыванию протокола IPv6.

Рекомендации по развертыванию протокола IPv6 в корпоративных сетях, RFC 7381[6]

Этот документ предлагает архитекторам и администраторам корпоративных сетей возможные стратегии и рекомендации по развертыванию протокола IPv6. Общая задача внедрения заключается в том, чтобы обеспечить услуги доступа в Интернет через IPv6 и поддержку этого протокола внутри корпоративной IT-сети, продолжая при этом поддерживать IPv4. В результате общего перехода большинство сетей перейдут из среды IPv4 в сетевую среду с двойным стеком и, в конечном итоге, в режим работы исключительно с IPv6. Документ рассматривает различные фазы развертывания и останавливается на таких аспектах, как адресация, маршрутизация, безопасность и мониторинг.

Рекомендации по предотвращению IP-спуфинга

Решение проблемы IP-спуфинга, когда IP-адреса отправителя пакетов подменяются на адрес вне сети отправителя (в случае намеренной рефлекторной атаки – обычно на адрес жертвы), было впервые предложено еще в 1998 году в RFC2267, который через два года был обновлен и получил категорию передовой практики BCP38[7]. Суть решения заключается в установке на граничных маршрутизаторах, к которым подключаются клиенты, «входных фильтров» (ingress filters). Эти фильтры разрешают только трафик, отправленный с исходных адресов блока клиента, и запрещают злоумышленнику использовать «недействительные» исходные адреса, находящиеся за пределами этого диапазона префиксов. Документ не содержит конкретных примеров конфигурации. Этот вопрос более детально рассматривается в документе «Входная фильтрация для сетей, использующих несколько провайдеров», RFC3704[8], также известном как передовая практика BCP84. В документе изложены пять различных способов реализации входных фильтров, начиная от статических листов доступа и заканчивая вариантами технологии переадресации обратного пути (точнее – Unicast Reverse Path Forwarding, uRPF). Применение технологий uRPF включает риски, связанные с возможной асимметрией трафика, например, когда клиент отправляет трафик через одного

---

6 RFC 7318: Enterprise IPv6 Deployment Guidelines,
URL: https://www.rfc-editor.org/rfc/rfc7318

7 BCP 38: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing,
URL: https://www.rfc-editor.org/info/bcp38

8 RFC 3704: Ingress Filtering for Multihomed Networks,
URL: https://www.rfc-editor.org/rfc/rfc3704

провайдера, а предпочитает получать трафик через другого. BCP84 содержит рекомендации по применению конкретного варианта uRPF в зависимости от сетевой топологии. Примечательно, что фильтрация на основе статических листов доступа до сих пор считается наиболее надежной. Ее основным недостатком является плохая масштабируемость и необходимость дополнительной автоматизации построения фильтров.

Конкретные примеры использования описанных подходов в конфигурациях для различного сетевого оборудования приведены в «Практическом руководстве рабочей группы по борьбе со спуфингом», RIPE-431[9].

Более широкий спектр оборудования и соответствующие примеры конфигурации можно также найти в описании каждого участника программы MANRS для производителей сетевого оборудования[10].

Конфигурация префикса IPv6 для конечных пользователей, RIPE-690[11]

Этот документ описывает передовую практику при присвоении IPv6-префикса конечным пользователям. Под конечными пользователями подразумеваются как домашние сети, так и корпоративные. Как отмечается в документе, неправильный выбор при проектировании сети IPv6 рано или поздно приведет к негативным последствиям для развертывания и потребует дальнейших усилий, таких как изменение нумерации, когда сеть уже работает. В этом плане IPv6 существенно отличается от IPv4 и требует другого подхода при проектировании адресного плана. Документ, в частности, рекомендует минимальный размер префикса /48, позволяя /56 для домашних сетей.

Вопросы эксплуатационной безопасности сетей IPv6, RFC 9099[12]

В этом документе анализируются проблемы эксплуатационной безопасности, связанные с несколькими типами IPv6-сетей, и предлагаются технические и процедурные методы их решения. Как отмечается в документе, знания и опыт безопасной эксплуатации сетей IPv4 доступны независимо от того, является ли оператор интернет-провайдером (ISP) или внутренней сетью предприятия. Однако IPv6 создает некоторые новые проблемы безопасности. Данный документ рассчитан на сетевых администраторов, предлагая необходимые практические рекомендации, ориентированные на эксплуатацию,

9 RIPE Anti-Spoofing Task Force HOW-TO, https://www.ripe.net/publications/docs/ripe-43

10 https://www.manrs.org/equipment-vendors/participants/

11 Best Current Operational Practice for Operators: IPv6 prefix assignment for end-users - persistent vs non-persistent, and what size to choose, https://www.ripe.net/publications/docs/ripe-690

12 RFC 9099: Operational Security Considerations for IPv6 Networks, URL: https://www.rfc-editor.org/rfc/rfc9099

и анализируя преимущества и недостатки определенных вариантов решений. Документ охватывает широкий спектр вопросов – от адресации и маршрутизации, до использования переходных технологий, мониторинга и защищенности устройств. Отдельно выделены соображения безопасности для корпоративных и домашних сетей и сетей сервис-провайдера.

Требования по поддержке IPv6 в ИКТ-оборудовании, RIPE-772[13]

Для обеспечения плавного и экономически эффективного внедрения IPv6 в компьютерных сетях важно, чтобы крупные коммерческие, государственные или исследовательские и образовательные предприятия при составлении тендеров на оборудование и поддержку информационных и коммуникационных технологий (ИКТ) правильно формулировали требования к поддержке протокола IPv6. Основная проблема заключается в том, что функциональность IPv6 определена во множестве стандартов и технических спецификаций. Кроме того, для различных типов устройств определенная функциональность может быть как обязательной, так и дополнительной. Таким образом, точная формулировка требований требует знания конкретных стандартов/спецификаций и их применимости к конкретному типу оборудования.

Этот документ предлагает передовую практику для поддержки организаций в таких тендерных процессах. Документ также содержит рекомендации для разработчиков программного обеспечения и системных интеграторов.

Маршрутизация

В этом разделе мы кратко остановимся на практиках и рекомендациях по внедрению и эксплуатации системы междоменной маршрутизации с особым фокусом на безопасность.

Управление и безопасность BGP, BCP 194[14]

Этот документ является всесторонним обзором мер по защите BGP. В нем описаны меры по защите сеансов BGP, такие как время жизни (TTL), опция аутентификации TCP (TCP-AO) и фильтрация в плоскости управления. В нем также описываются меры по более строгому контролю маршрутной информации с использованием префиксной фильтрации и автоматизации префиксных фильтров (как с помощью IRR, так и системы RPKI), фильтрации по максимальным префиксам, фильтрации путей автономной системы (AS), подавления колебаний маршрутов и очистки сообществ BGP.

---

[13] Requirements For IPv6 in ICT Equipment, https://www.ripe.net/publications/docs/ripe-772

[14] BCP 194: BGP Operations and Security, URL: https://www.rfc-editor.org/info/bcp194

Требования к доверяющим сторонам инфраструктуры открытых ключей ресурсов (RPKI), RFC 8897[15]

Документ суммирует все требования к программному обеспечению пользователей, использующих инфраструктуру открытых ключей ресурсов (RPKI) для проверки аутентичности хранящейся в ней информации, т.н. доверяющие стороны. В документе приводятся требования, которые присутствуют в более чем десяти RFC, что облегчает разработчикам ознакомление с этими требованиями.

Требования разбиты на четыре группы:

- Извлечение и кеширование объектов репозитория RPKI.
- Обработка сертификатов и списков отзыва сертификатов (CRL).
- Обработка подписанных объектов репозитория RPKI.
- Распространение проверенного кеша данных RPKI.

Предполагается, что документ будет обновляться с учетом новых или измененных требований по мере обновления этих RFC или написания дополнительных RFC.

Требования к эксплуатационной безопасности для инфраструктуры IP-сети крупного интернет-провайдера (ISP), RFC 3871[16]

Этот документ определяет список эксплуатационных требований безопасности для инфраструктуры IP-сетей крупных интернет-провайдеров (ISP) - маршрутизаторов и коммутаторов. В документе используется подход, позволяющий определять «профили», которые представляют собой набор требований, применимых к определенным контекстам топологии сети (вся сеть, только ядро, только периферия...). Цель состоит в том, чтобы предоставить сетевым операторам четкий и краткий способ донести свои требования безопасности до поставщиков оборудования.

Практическое использование языка спецификации политик маршрутизации (RPSL), RFC 2650[17]

RPSL используется для описания политики маршрутизации интернет-провайдера и создания соответствующих объектов в реестре маршрутизации Интернета (IRR). Данный документ представляет собой руководство по использованию языка RPSL. В нем изложено, как указывать различные политики и конфигурации

---

15 RFC 8897: Requirements for Resource Public Key Infrastructure (RPKI) Relying Parties,
URL: https://www.rfc-editor.org/rfc/rfc8897

16 RFC 3871: Operational Security Requirements for Large Internet Service Provider (ISP) IP Network Infrastructure,
URL: https://www.rfc-editor.org/rfc/rfc3871

17 RFC 2650: Using RPSL in Practice,
URL: https://www.rfc-editor.org/rfc/rfc2650

маршрутизации с помощью RPSL, как регистрировать эти политики в IRR и как анализировать их с помощью инструментов анализа политик маршрутизации, например, для создания конфигураций маршрутизатора для конкретного типа оборудования.

Взаимосогласованные нормы безопасности маршрутизации, MANRS[18]

MANRS является глобальной инициативой, запущенной в 2014 году сетевыми операторами при поддержке Internet Society. Начавшись с программы для интернет-провайдеров, на сегодняшний день MANRS поддерживает отдельные программы для операторов точек обмена трафиком, операторов CDN и облачных услуг, а также производителей сетевого оборудования. По состоянию на конец 2023 года более тысячи операторов различных типов являлись участниками инициативы.

Для каждой категории операторов MANRS определяет набор требований (Actions), которым оператор должен удовлетворять для участия в инициативе. Так, например, сетевые операторы должны продемонстрировать выполнение следующих требований:

- Предотвращение передачи ложных анонсов. Интернет-провайдер должен обеспечить фильтрацию анонсов от своих клиентов и собственных сетей для избежания атак захвата или утечки маршрута.
- Противодействие IP-спуфингу. Системы провайдера должны использовать соответствующие фильтры, блокирующие трафик клиентов с подменой адреса отправителя.
- Поддержка коммуникации и координации в глобальном масштабе. Поскольку реакция на атаки и операционная координация являются необходимыми факторами эффективного противодействия, данное требование устанавливает публичную доступность контактной информации оператора.
- Поддержка маршрутной информации в глобальном масштабе. Это требование предусматривает поддержку актуальной информации о маршрутах провайдера в соответствующих регистратурах – IRR и RPKI.

Необходимо отметить, что инициатива MANRS не определяет новых требований или практик. Она базируется на существующих подходах решения проблем и фокусируется на конечном результате.

[18] Mutually Agreed Norms for Routing Security (MANRS), https://www.manrs.org

Глоссарий

Термин/ аббревиатура/ название организации	Название, принятое в русском языке	Определение
AfriNIC, African Network Information Centre	AfriNIC	Региональная интернет-регистратура, действует в странах Африки и Индийского океана.
ALAC, At-Large Advisory Committee	Расширенный консультативный комитет	Комитет в структуре ICANN, представляющий интересы сообщества индивидуальных интернет-пользователей.
Anycast	Аникаст	Технология аникаст предполагает, что один и тот же IP-адрес используется несколькими хостами, расположенными в различных сетях. В случае использования аникаст в Интернете в результате процесса выбора лучшего пути BGP трафик направляется к ближайшему относительно отправителя аникаст-узлу. Эта технология нашла широкое применение в DNS для повышения устойчивости и производительности системы. Так, например, большинство серверов корневой зоны используют аникаст.
APNIC, Asia-Pacific Network Information Centre	APNIC	Региональная интернет-регистратура, действует в странах Азиатско-Тихоокеанского региона.
APTLD, Asia Pacific Top Level Domain Association	APTLD, Азиатско-тихоокеанская ассоциация доменов верхнего уровня	Ассоциация регистратур национальных доменов стран Азиатско-Тихоокеанского региона.
ARIN, American Registry for Internet Numbers,	ARIN	Региональная интернет-регистратура региона Северной Америки, действует в США, Канаде, на многих Карибских и Североатлантических островах.
ARPANET	ARPANET	ARPANET была первой глобальной сетью коммутации пакетов с распределенным управлением и одной из первых компьютерных сетей, реализовавших набор протоколов TCP/IP. Обе технологии стали технической основой Интернета. ARPANET была создана Агентством перспективных исследовательских проектов (ARPA) Министерства обороны США.

Термин/ аббревиатура/ название организации	Название, принятое в русском языке	Определение
AS, Autonomous System	Автономная система	Система IP-сетей, управляемых одним или несколькими операторами, имеющими единую политику маршрутизации в Интернете. Автономные системы обмениваются друг с другом информацией о маршрутах с помощью протокола BGP для построения общей «карты» Интернета.
ASCII, American Standard Code for Information Interchange	Кодировка ASCII	Кодировка, основанная на английском алфавите, в которой используются цифровые соответствия для букв от a до z, цифр от 0 до 9 и знака «дефис». До декабря 2007 года оставалась наиболее часто используемой кодировкой в Интернете.
ASO, Address Supporting Organization	Организация поддержки адресов	Организация в структуре ICANN, представляющая региональных интернет-регистраторов.
Authoritative Server DNS	Авторитетный сервер DNS	Сервер DNS, обслуживающий определенную зону DNS, связанную с доменным именем. Ответы этих серверов отражают информацию зоны, как ее опубликовал владелец домена.
BGP, Border Gateway Protocol	Протокол граничного шлюза	BGP — это основной протокол динамической маршрутизации в Интернете. BGP обеспечивает обмен маршрутами между сетями (автономными системами) для построения динамической «карты» глобальной связности каждой сетью.
Bluetooth	Bluetooth	Протокол беспроводной связи, использующийся в системах управления и обмена информацией с различными гаджетами — от фитнес-сенсоров до наушников и спикеров. Новая версия протокола — Bluetooth Low-Energy (BLE), или Bluetooth Smart — является важным протоколом для приложений IoT.
Botnet	Ботнет	Ботнет, сокращенное от «сеть (ро)ботов», это сеть компьютеров, модемов, «умных вещей» и других устройств, зараженных вредоносным ПО и находящихся под контролем атакующей стороны. Каждая отдельная машина в такой сети называется ботом.
CA, Certificate Authority	Удостоверяющий центр	Удостоверяющий центр (УЦ) — это организация, которая хранит, подписывает и выдает цифровые сертификаты. УЦ является важным элементом инфраструктуры открытых ключей, основанной на асимметричной криптографии. Цифровой сертификат удостоверяет владение открытым ключом названным субъектом сертификата. Это позволяет другим (проверяющим сторонам) полагаться на подписи или другие утверждения с использованием секретного ключа, который соответствует сертифицированному открытому ключу.
Cache	Кеш	Аппаратный или программный компонент, который хранит данные для ускорения обслуживания будущих запросов на эти данные. Данные, хранящиеся в кеше, могут быть копией данных, хранящихся в другом месте.

Термин/ аббревиатура/ название организации	Название, принятое в русском языке	Определение
ccNSO, Country Code Names Supporting Organization	Организация поддержки национальных доменов	Организация в структуре ICANN, представляющая интересы национальных доменов.
ccTLD, Country Code Top Level Domain, Country Code	Национальный домен верхнего уровня	Домен верхнего уровня Интернета, обычно используемый или зарезервированный для страны, суверенного государства или зависимой территории, идентифицированной кодом страны. Все идентификаторы ASCII ccTLD состоят из двух букв.
CDN, Content Delivery Network	Сети доставки контента	Географически распределенная сеть прокси-серверов, кеширующих данные, запрашиваемые пользователями. Благодаря этому контент располагается ближе к пользователю, улучшая производительность и надежность, а также позволяя оптимизировать потоки трафика.
CENTR, Council of European National TLD Registries	Ассоциация регистратур национальных доменов европейских стран CENTR	Совет европейских национальных регистратур доменов верхнего уровня/Ассоциация регистратур национальных доменов европейских стран.
CIDR, Classless Inter-Domain Routing	Бесклассовая междоменная маршрутизация	IETF стандартизировал CIDR в 1993 году, чтобы заменить предыдущую классовую архитектуру сетевой адресации в Интернете. Цель CIDR заключалась в том, чтобы замедлить рост таблиц маршрутизации на маршрутизаторах в Интернете и помочь замедлить быстрое исчерпание адресов IPv4. В рамках CIDR размер сетевой части IP-адреса не является фиксированным и определяется параметром «размер префикса» (RFC 1518, RFC 1519).
Cybersquatting	Кибersквоттинг	Регистрация доменных имен, совпадающих или содержащих в своем названии наименования защищенных торговых марок и товарных знаков, лицами либо организациями, которые не являются их правообладателями. Также – регистрация доменных имен, сходных или совпадающих с доменами сайтов известных компаний, организаций, выдающихся личностей и т.д.
DANE, DNS-Based Authentication of Named Entities	Протокол аутентификации поименованных объектов с использованием DNS	DANE позволяет владельцу домена указать в DNS, какой сертификат TLS/SSL должно использовать приложение или служба для подключения к сайту с этим доменным именем (RFC 6698, RFC 7671).
DARPA, Defense Advanced Research Projects Agency	Управление перспективных исследовательских проектов Министерства обороны США	Управление Министерства обороны США, отвечающее за разработку новых технологий для использования в интересах вооружённых сил. DARPA отвечало за финансирование разработки университетами распределённой компьютерной сети ARPANET, являвшейся предтечей Интернета.

Термин/ аббревиатура/ название организации	Название, принятое в русском языке	Определение
DDoS, Distributed Denial-of-Service Attack	Распределенная атака типа «отказ в обслуживании»	Распределенная атака типа «отказ в обслуживании» — это злонамеренная попытка нарушить нормальный трафик целевого сервера, службы или сети путем перегрузки цели или окружающей ее инфраструктуры потоком интернет-трафика. Атакующие системы распределены географически, что усложняет противодействие.
DHCP, Dynamic Host Configuration Protocol	Протокол динамической конфигурации хоста	DHCP — протокол управления сетью, используемый для динамического назначения IP-адреса любому устройству или узлу в сети (RFC 2131).
DKIM, DomainKeys Identified Mail	Идентификация почты с помощью DomainKey	Протокол DKIM использует асимметричную криптографию для защиты с помощью электронной подписи полей сообщения электронной почты, таких как «От:», «Кому:», «Дата:» (RFC 6376).
DMARC, Domain-based Message Authentication, Reporting and Conformance	Аутентификация, отчетность и определение соответствия сообщений на основе доменного имени	Протокол, позволяющий владельцу домена публиковать политики обработки сообщений для получателей электронной почты, исходящей из этого домена, и запрашивать отчеты об аутентификации полученной почты (RFC 7489).
DNS Abuse	Противоправное использование системы DNS	Выделяют 5 общих категорий DNS Abuse: распространение вредоносного ПО, создание бот-сетей, фишинг, фарминг (распространение специального ПО, тайно подменяющего данные и перенаправляющего пользователей на вредоносные страницы вместо легитимных) и спам (в случае, если спам-сообщения используются для других форм DNS Abuse – распространения вредоносного ПО, фишинговых ссылок и т.д.).
DNS Resolution	Разрешение доменных имен	Процесс преобразования имен доменов в IP-адреса.
DNS Resolver	DNS-резолвер	Сервер, кеширующий информацию об IP-адресах различных веб-ресурсов. Именно на него направляется DNS-запрос с пользовательского устройства. В большинстве случаев это сервер провайдера, предоставляющего пользователю услуги доступа в Интернет. В случае отсутствия необходимой информации в кеше резолвер направляет запрос на авторитетный DNS-сервер.
DNS, Domain Name System	Система доменных имен, система DNS	Служит для преобразования доменных имен в IP-адреса и обратного процесса, тем самым обеспечивая функционирование всей глобальной сети.
DNSSEC, Domain Name System Security Extensions	DNSSEC	Набор расширений протокола DNS, который обеспечивает криптографическую защиту DNS-запросов. Играет важную роль в обеспечении безопасности системы.

Термин/ аббревиатура/ название организации	Название, принятое в русском языке	Определение
DoH, DNS over HTTPS	DNS поверх HTTPS	Протокол для выполнения трансляции DNS по протоколу HTTPS, который используется для шифрования данных, передаваемых между клиентом и севером DNS. Целью этого метода является повышение конфиденциальности и безопасности пользователей путём предотвращения перехвата и манипулирования данными DNS с помощью атак типа «Атака посредника» (RFC 8484).
Domain name	Доменное имя	Доменное имя — символьное имя, служащее для идентификации областей общего пространства имен DNS, которые являются единицами административной автономии. Каждая из таких областей называется доменом. Домены организованы в иерархическую структуру.
Domain zone	Доменная зона	Доменная зона - часть общего пространства DNS, содержащая информацию, связанную с конкретным доменным именем. Так, например, в доменной зоне указаны имена авторитетных серверов DNS, почтовых серверов, а также поддоменов.
DoQ, DNS over QUICK	DNS поверх QUICK	Протокол для выполнения трансляции DNS по протоколу QUICK, который используется для шифрования данных, передаваемых между клиентом и севером DNS. Целью этого метода является повышение конфиденциальности и безопасности пользователей путём предотвращения перехвата и манипулирования данными DNS с помощью атак типа «Атака посредника» (RFC 9250).
DoT, DNS over TLS	DNS поверх TLS	Протокол для выполнения трансляции DNS по протоколу TLS, который используется для шифрования данных, передаваемых между клиентом и севером DNS. Целью этого метода является повышение конфиденциальности и безопасности пользователей путём предотвращения перехвата и манипулирования данными DNS с помощью атак типа «Атака посредника» (RFC 7858).
ENOG, Eurasia Network Operators' Group	ENOG – Евразийский форум интернет-операторов	Группа сетевых операторов Евразии – форум специалистов сетевых технологий стран Восточной Европы, Кавказского региона и Средней Азии.
FTP, File Transfer Protocol	Протокол передачи файлов	FTP — протокол передачи файлов от сервера к клиенту, появившийся в 1971 году задолго до HTTP и даже до TCP/IP, благодаря чему является одним из старейших прикладных протоколов (RFC 959).
GAC, Goverment Advisory Committee	Правительственный консультативный комитет	Комитет в структуре ICANN, представляющий интересы национальных правительств и государственных организаций.
gNSO, Generic Names Supporting Organization	Организации поддержки общих доменов верхнего уровня	Организация в структуре ICANN, основной орган управления общими доменами верхнего уровня.

Термин/ аббревиатура/ название организации	Название, принятое в русском языке	Определение
gTLD, generic Top Level Domain	Общий домен верхнего уровня	Домен верхнего уровня Интернета, созданный для сегментации адресов электронных ресурсов определённого класса организаций либо сообществ.
HTTP, HyperText Transfer Protocol	Протокол передачи гипертекста	HTTP — сетевой протокол прикладного уровня, который изначально предназначался для получения с серверов гипертекстовых документов в формате HTML, а с течением времени стал универсальным средством взаимодействия между узлами как Всемирной паутины, так и изолированных веб-инфраструктур.
IANA, Internet Assigned Numbers Authority	Администрация номерного пространства Интернета	Организация, ответственная за регистратуры цифровых идентификаторов, включая доменные имена верхнего уровня, параметры протоколов, IP-адреса и номера автономных систем.
ICANN, Internet Corporation for Assigned Names and Numbers	ICANN, Корпорация по управлению доменными именами и IP-адресами	Независимая международная некоммерческая частная организация, регулирующая вопросы, связанные с доменными именами, IP-адресами и другими аспектами функционирования Интернета.
IDN, Internationalized Domain Name	Интернационализированный домен, домен верхнего уровня с использованием национального алфавита	Домен, использующий символы алфавитов, отличных от латинского (например, домен .рф). Все имена в интернационализированных доменах могут состоять исключительно из символов этих алфавитов.
IEEE, Institute of Electrical and Electronics Engineers	Институт инженеров по электротехнике и электронике	Международная некоммерческая ассоциация специалистов в области техники, мировой лидер в области разработки стандартов. Стандарты IEEE работают в различных отраслях промышленности, включая энергетику, здравоохранение, информационные и телекоммуникационные технологии (ИКТ), транспорт и многие другие. В области ИКТ наиболее значительные стандарты, имеющие отношение к Интернету, разработаны в комитете IEEE 802 LAN/ MAN Standards Committee (LMSC).
IETF, Internet Engineering Task Force	Инженерный совет Интернета	Международное сообщество проектировщиков, учёных, сетевых операторов и провайдеров, занятое развитием протоколов и архитектуры интернета. Входит в состав ISOC.
IoT, Internet of Things	Интернет вещей	Интернет вещей - сеть обмена данными между физическими объектами («вещами»), оснащёнными встроенными средствами и технологиями для взаимодействия друг с другом или с внешней средой.
IP-spoofing	IP-спуфинг	Фальсификация адреса отправителя. Этот метод обычно используется в атаках отражения, или рефлекторных атаках, когда адрес отправителя заменяется адресом жертвы. При этом обратный трафик, например, ответы на запросы, будет направлен не отправителю, а жертве, вызывая перегрузку сети или приложения.

Термин/ аббревиатура/ название организации	Название, принятое в русском языке	Определение
IRR, Internet Routing Registry	Интернет-регистратуры маршрутизации	Общедоступные базы данных для регистрации сетевыми операторами политики маршрутизации и дополнительной информации. Эта информация используется другими операторами для проверки правильности полученных анонсов.
ISOC, Internet Society	Общество Интернета	Международная профессиональная организация, занимающаяся развитием и обеспечением доступности сети Интернет.
ITU, International Telecommunication Union	МСЭ, Международный союз электросвязи	Международная организация в структуре ООН, формулирующая рекомендации в области телекоммуникаций и радио, а также регулирующая вопросы международного использования радиочастот.
KSK, Key Signing Key	Ключ для подписи ключей	Помимо ключа подписи зоны, серверы имен DNSSEC также имеют ключ подписи ключа (KSK). KSK защищает запись DNSKEY или, другими словами, ключ подписи зоны ZSK.
LACNIC, Latin American and Caribbean Network Information Centre	LACNIC	Региональная интернет-регистратура, действует в странах Латинской Америки и Карибского бассейна.
Legacy TLD	“Старые” домены	Термин, используемый для обозначения общих доменов верхнего уровня, которые были запущены до 2012 года. Может переводиться как «старые» домены. Подразумевает прежде всего доменные зоны .com, .net, и org.
Multistakeholderism	Мультистейкхолдеризм, принцип многостороннего управления Интернетом	Принцип многостороннего управления Интернетом, основанный на равноправном участии всех заинтересованных сторон (stakeholders). За неимением адекватного русского аналога используется калька «мультистейкхолдеризм». Принцип подразумевает, что в выработке всех ключевых решении участвуют группы, представляющие интересы всех сторон: правительств, международных организаций бизнеса, научно-технического сообщества и пользователей. Сами же решения принимаются на основе консенсуса всех групп.
NAT, Network Address Translation	Сетевые трансляторы пртоколов	Устройства NAT производят отображение (трансляцию) одного пространства IP-адресов в другое путем изменения информации о сетевых адресах в IP-заголовке пакетов.
New gTLD, new generic Top Level Domain	Новый общий домен верхнего уровня	Домен, запущенный в рамках программы новых общих доменов верхнего уровня корпорации ICANN, начиная с 2012 года.
NRO, Number Resource Organisation	Организация номерных ресурсов	Организация, обеспечивающая координацию между региональными интернет-регистратурами.
NSFNET, National Science Foundation Network	Компьютерная сеть Национального фонда науки США	NSFNET — компьютерная сеть Национального фонда науки США, образованная в 1984 году и служившая опорной сетью Интернета в начале 1990-х годов.

Термин/ аббревиатура/ название организации	Название, принятое в русском языке	Определение
OpenFLow	OpenFLow	Протокол, реализующий концепцию SDN. Протокол разрабатывается и поддерживается Фондом открытых сетей (Open Networking Foundation, ONF).
Packet switched networks	Сети коммутации пакетов	Сети, в которых данные передаются в виде блоков конечного размера — пакетов. Пакет данных помимо полезной нагрузки содержит адрес получателя и другую информацию, необходимую для его обработки сетью или оконечным устройством. Обычно каждый пакет обрабатывается сетью независимо. Эта технология коренным образом отличается от другой концепции — переключения каналов, при которой ресурсы сети от отправителя до получателя резервируются, создавая канал с определенными характеристиками.
Peer-to-peer (P2P) networks	Одноранговые сети	Одноранговая или пиринговая сеть — оверлейная компьютерная сеть, основанная на равноправии участников. Часто в такой сети отсутствуют выделенные серверы, а каждый узел (peer) является как клиентом, так и выполняет функции сервера.
Peering	Пиринг	Тип взаимодействия между сетями с целью обмена трафиком между собой, включая всех собственных клиентов. Обычно ни одна из сторон не платит другой за трафик, вместо этого каждый получает и удерживает доход от своих клиентов. Другим типом взаимодействия является «транзит».
PKI, Public Key Infratsructure	Инфраструтура открытых ключей	Инфраструктура открытых ключей (PKI) — это сочетание политик, процедур и технологий, необходимых для управления цифровыми сертификатами в схеме шифрования с открытым ключом. Цифровой сертификат — это электронная структура данных, которая связывает объект, будь то учреждение, физическое лицо, компьютерная программа, доменное имя, с его открытым ключом. Цифровые сертификаты используются для защиты связи с использованием асимметричной криптографии.
Punycode	Кодировка Punycode	Кодировка, обеспечивающая совместимость интернационализированных доменов с традиционными для системы DNS доменами, использующими кодировку ASCII.
RDAP, Registration Data Access Protocol	RDAP	Протокол, который рассматривается как замена WHOIS после принятия Общего регламента по защите данных. В отличие от WHOIS накладывает серьезные ограничения на доступ к персональным данным регистрантов.
Reflection attacks	Атаки отражения/ рефлекторные атаки	Эти атаки используют возможность фальсификации адреса отправителя, когда адрес отправителя заменяется адресом жертвы. При этом обратный трафик, например, ответы на запросы, будет направлен не отправителю, а жертве, вызывая перегрузку сети или приложения.

Термин/ аббревиатура/ название организации	Название, принятое в русском языке	Определение
RFC, Request for Comments	Рабочее предложение	Документ из серии пронумерованных информационных документов IETF, которые содержат технические рекомендации, спецификации и стандарты, применяемые в глобальной сети и регламентирующие ее деятельность.
RIPE NCC, Reseaux IP Europeens Network Coordination Centre/RIPE Network Coordination Centre	RIPE NCC, Координационный центр распределения ресурсов сети Интернет в Европейском регионе	Региональная интернет-регистратура региона Европейского региона, действует в Европе, странах Средней Азии и Ближнего Востока.
RIR, Regional Internet Registry, также LIR, Local Internet Registry	Региональная интернет-регистратура	Организация, занимающаяся технической координацией функционирования Интернета: выделением IP-адресов и номеров автономных систем.
ROA, Route Object Authorisation	Авторизация объекта маршрута	ROA — это объект RPKI с криптографической подписью, в котором указывается, какая автономная система (AS) имеет право быть источником анонсирования определенного IP-префикса или набора префиксов.
Root Server, также Root Name Server	Корневой сервер DNS	Корневые серверы обеспечивают работу корневой зоны. Хранят списки всех DNS-серверов доменов верхнего уровня. На сегодня в мире существует 13 корневых серверов, обозначенных буквами латинского алфавита от a до m. Благодаря используемой технологии аникаст, число реальных серверов на конец 2023 года составило 1750.
Root Zone	Корневая зона DNS	Часть DNS, содержащая информацию обо всех доменах верхнего уровня и их серверах. Обслуживается 13 корневыми серверами. Благодаря используемой технологии аникаст, число реальных серверов на конец 2023 года составило 1750.
Routing	Маршрутизация трафика	Маршрутизация — это процесс выбора пути для трафика между отправителем и получателем.
RPKI, Resourve Public Key Infrastructure	Инфраструтура открытых ключей для интернет-ресурсов	RPKI является специализированной инфраструктурой открытых ключей, удостоверяющей связь между конкретными блоками IP-адресов или номерами AS и владельцами этих номерных ресурсов Интернета. Сертификаты являются доказательством права владельца ресурса на использование своих ресурсов.
RRL, Response Rate limiting	Ограничение частоты ответов	Этот подход предусматривает ограничение частоты запросов с идентичным ответом от одного и того же клиента.

Термин/ аббревиатура/ название организации	Название, принятое в русском языке	Определение
SDN, Software-Defined Networking	Программно-определяемая сеть	SDN — это подход к организации сети, в котором используются программные контроллеры или интерфейсы прикладного программирования (API) для связи с базовой аппаратной инфраструктурой и маршрутизации трафика в сети.
SPF, Sender Policy Framework	Система политики отправителя	Стандартизированный метод предотвращения подделки адреса отправителя. SPF позволяет администраторам указать, каким хостам разрешено отправлять почту от имени данного домена, путем создания специальной записи SPF в DNS (RFC 7208).
SSAC, Security and Stability Advisory Committee	Консультативный комитет по безопасности и стабильности	Комитет в структуре ICANN, предоставляет рекомендации по вопросам, связанным с безопасностью и целостностью системы распределения имен и адресов Интернета.
sTLD, Sponsored TLD	Спонсируемый общий домен верхнего уровня	К этой категории относятся домены, делегированные по заявкам различных организаций либо коммерческих компаний – например, .asia, .coop, .museum. С началом программы новых общих доменов верхнего уровня практика делегирования спонсируемых доменов была прекращена ICANN.
TCP/IP, Transmission Control Protocol/Internet Protocol	Протокол управления передачей/Интернет-протокол	TCP/IP означает и представляет собой набор протоколов связи, используемых для соединения сетевых устройств в Интернете. TCP/IP имеет четыре уровня абстракции: канальный уровень, уровень IP, транспортный уровень и уровень приложений (RFC 1122).
TLS, Transport Layer Security	Безопасность транстпортного уровня	TLS — это криптографический протокол, предназначенный для защиты передаваемых данных в компьютерной сети. Протокол широко используется в таких приложениях, как доступ к веб-сайтам (HTTPS), электронная почта, обмен мгновенными сообщениями и передача голоса по IP. TLS пришел на смену протоколу SSL.
Transit	Транзит	Тип взаимоотношения между сетями, при котором одна из сетей предоставляет другой услугу доступа к глобальному Интернету. Как правило, транзит является платной услугой, стоимость которой зависит от типа подключения и объема передаваемого трафика. Другим типом взаимодействия является "пиринг".
TSIG, Transaction SIGnature	Подпись транзакции	Протокол, обеспечивающий аутентификацию и проверку целостности обновлений DNS-зоны. Его можно использовать для аутентификации динамических обновлений как поступающих от утвержденного клиента. Он также используется для защищенной передачи данных между основным и вторичным серверами DNS (RFC 2845).
Typosquatting	Тайпсквоттинг	Регистрация доменных имен, сходных в написании с доменами популярных сайтов. В дальнейшем подобные домены обычно используются для мошенничества, создания фишинговых страниц и т.д.

Термин/ аббревиатура/ название организации	Название, принятое в русском языке	Определение
Unicode	Юникод	Стандарт кодирования символов национальных алфавитов, разработанный консорциумом «Юникод».
Universal Acceptance	Универсальное принятие	Проблема единообразного принятия, проверки, хранения, обработки и отображения во всех приложениях и интернет-сервисах всех допустимых доменных имен и адресов электронной почты, включая интернационализированные.
uRPF, Unicast Reverse Path Forwarding	Юникастовая передача по обратному пути	Данный поход ограничивает вредоносный трафик в сети. Эта функция позволяет устройствам проверять доступность адреса источника в пересылаемых пакетах и ограничивать появление поддельных или искаженных адресов в сети. Если IP-адрес отправителя недействителен, функция uRPF отбрасывает пакет.
W3C, World Wide Web Consortium	Консорциум Всемирной паутины	Организация, разрабатывающая и внедряющая технологические стандарты для сети Интернет.
WHOIS	WHOIS	Сетевой протокол, используемый для записи регистрационных данных владельцев IP-адресов и регистрантов доменных имен и их контактной информации. Принятие Евросоюзом Общего регламента по защите данных (GDPR) сделало практически невозможным использование WHOIS в его прежнем виде, поскольку он открывал доступ к информации о регистрантах, которая относится к категории персональных данных.
WIPO, World Intellectual Property Organization	ВОИС, Всемирная организация интеллектуальной собственности	Международная организация, контролирующая соблюдение ключевых международных конвенций в области интеллектуальной собственности. Ее арбитражный центр/центр по арбитражу и посредничеству является одной из основных мировых инстанций, рассматривающих доменные споры в рамках UDRP.
ZigBee	ZigBee	ZigBee — один из популярных протоколов, применяющихся в системах домашней и офисной автоматизации, например, в системах освещения.
ZSK, Zone Signing Key	Ключ для подписи зоны	Каждая зона в DNSSEC имеет ключ для подписи зоны (ZSK). Ключ состоит из двух частей: частная часть ключа ставит цифровую подпись для каждого набора RRset в зоне, а общедоступная часть используется для проверки подписи и, соответственно, целостности полученных данных.

Для заметок

Робачевский Андрей
ИНТЕРНЕТ ИЗНУТРИ
Архитектура экосистемы Интернета

Дизайн, компьютерная верстка Ивлянов Дмитрий
Корректор Рябова Наталья

Подписано в печать: 17.05.2024
Формат 70x100/32.
Объем 15 печ.л. Тираж 300 экз.

Заказ №2573
Отпечатано с готового оригинал-макета заказчика
в типографии ООО «Артик Принт»
Тел.: +7 (495) 646 49 00, www.serpantin.agency, info@serpantin.agency

Андрей Робачевский — ветеран Интернета и эксперт в области интернет-технологий и безопасности. Он начал работать в сфере Интернета с начала 1990-х гг. в составе команды по созданию федеральной университетской компьютерной сети России RUNNet. С 2002 по 2011 г. занимал должность технического директора региональной интернет-регистратуры RIPE NCC.

Под руководством Андрея Робачевского было осуществлено внедрение DNSSEC в обратных зонах DNS, а также построена разветвленная сеть узлов корневого DNS-сервера K-root. Его последующая деятельность в Internet Society и Global Cyber Alliance направлена на улучшение безопасности и стабильности инфраструктуры глобального Интернета на основе согласованных усилий и кооперации между участниками экосистемы Интернета. Он является одним из основателей инициативы MANRS (Mutually Agreed Norms for Routing Security).

Андрей также активно участвует в работе организации по разработке стандартов Интернета — IETF, с 2010 по 2012 г. он являлся членом Совета по архитектуре Интернета (IAB).

indata

ФОНД РАЗВИТИЯ СЕТЕВЫХ ТЕХНОЛОГИЙ «ИНДАТА»

Фонд развития сетевых технологий «ИнДата» образован в 2016 году. Направление деятельности Фонда: проведение научно-прикладных и аналитических исследований сетевой инфраструктуры сети Интернет (в первую очередь — его российской части), а также связанная с этим образовательно-просветительская деятельность.

Отечественному интернет-сообществу хорошо известны проекты Фонда — «Макроскопические исследования интернет-инфраструктуры» (ididb.ru), «Визуализация маршрутной информации» (DASPath.ru) и «Платформа измерений в сети Интернет». Цель проектов — поддержка целостности, устойчивости и безопасности Рунета.

Поиск и разработка новых опытных решений, осуществляемые Фондом «ИнДата», несомненно, являются вкладом в развитие цифровой экономики страны.

В рамках образовательно-просветительского направления Фонд осуществляет разносторонние коммуникации в профессиональном и академическом сообществе, направленные на распространение знаний и внедрение лучших практик в области управления Интернетом. Для этого Фонд не только успешно сотрудничает с ведущими университетами страны, участвуя в совместных образовательных и проектных программах, но и издаёт журнал «Интернет изнутри» (ii.org.ru), публикующий тематические технические, аналитические и обзорные статьи по актуальной проблематике интернет-индустрии.

Контакты:
+7 495 510-00-98
info@indata.org.ru
indata.org.ru

Андрей Робачевский

Интернет изнутри

Архитектура экосистемы Интернета

Отправить письмо, посмотреть репертуар кинотеатра, узнать последние новости, прокомментировать фото друзей, заказать билеты на самолет… Все это и многое другое сотни миллионов людей во всем мире делают каждый день, даже не задумываясь о том, что за совершением обыденных операций скрывается сложнейшая инфраструктура.

Как защитить ценную информацию в Сети? Как и куда передаются данные? Как создавались протоколы, технологии и связи, составляющие основу Всемирной паутины, и почему все так сложно, если все так просто?

Ответ на этот и многие другие вопросы дает «Интернет изнутри» — настоящий путеводитель по архитектуре Сети. Книга будет интересна сетевым операторам, разработчикам и администраторам интернет-услуг и приложений, а также всем тем, кто хочет больше узнать об интернет-технологиях.

ISBN 978-5-6052033-0-8