

Architectural Principles of the Internet

Архитектурные принципы Internet

Статус документа

Этот документ содержит информацию для сообщества Internet. Документ не задаёт каких-либо стандартов Internet и может распространяться без ограничений.

Аннотация

Сеть Internet и её архитектура развивались эволюционно от скромных систем без грандиозных планов (Grand Plan). Хотя эта эволюция является одной из основных причин успеха технологии, представляется полезным зафиксировать текущие принципы архитектуры Internet. Документ служит рекомендацией и представляет общий интерес, но никоим образом не задаёт формальной или неизменной эталонной модели.

Оглавление

1. Постоянное изменение.....	1
2. Существует ли архитектура Internet?.....	1
3. Общие проблемы проектирования.....	2
4. Проблемы имён и адресов.....	3
5. Сторонние проблемы.....	3
6. Конфиденциальность и проверка подлинности.....	3
Благодарности.....	3
Литература.....	3
Вопросы безопасности.....	4
Адрес редактора.....	4

1. Постоянное изменение

В поисках архитектурных принципов Internet необходимо помнить, что в сфере информационных технологий происходят постоянные изменения, которые отражаются в Internet. За 25 лет с момента создания ARPANET разные параметры Internet изменились от 1000 (скорость магистралей) до 1000000 (число хостов) раз. В такой среде некоторые принципы архитектуры неизбежно меняются. Принципы, которые несколько лет назад казались незыблемыми, сегодня уже устарели. Принципы, которые сегодня представляются священными, могут устареть завтра. Пожалуй единственным неизменным принципом остаётся изменчивость Internet.

Цель этого документа состоит не в том, чтобы задать догмы разработки протоколов Internet или даже сочетания этих протоколов. Документ скорее предназначен в качестве набора руководящих принципов, которые были сочтены полезными в прошлом и могут быть полезны при разработке новых протоколов или оценке разработок.

Хорошей аналогией развития Internet является постоянное обновление отдельных улиц и зданий в городе вместо полного разрушения и постройки нового города. Принципы архитектуры предназначены обеспечить основу для сотрудничества и разработки стандартов в виде небольшого «связующего набора правил» (spanning set), который создаст обширное, разнообразное и развивающееся пространство технологий.

Некоторые из современных импульсов к изменению включают пределы расширяемости IPv4, новые проблемы, вызванные применением гигабитных скоростей и технологий multimedia, а также необходимость обеспечивать качество обслуживания и гарантии безопасности в коммерческой среде Internet.

Лорд Kelvin заявлял в 1895 году: «Летательные аппараты тяжелее воздуха невозможны». Было бы глупо полагать, что приведённые ниже принципы являются чем-то большим, нежели просто картина текущего понимания.

2. Существует ли архитектура Internet?

- 2.1 В сообществе Internet многие считают, что архитектуры нет, а имеются лишь традиции, которые не были записаны в течение первых 25 лет (по крайней мере, не записаны IAB). Однако в более общем виде сообщество считает, что целью является связность, инструментом - протокол IP (Internet Protocol), а информация является сквозной, а не спрятанной где-то в сети.
- 2.2 Текущий экспоненциальный рост сети, кажется, показывает, что связность сама является «наградой» и она более ценна, чем любое отдельное приложение, такое как почта или WWW (World-Wide Web). Эта связность требует технического сотрудничества между сервис-провайдерами и процветает в условиях либерализации и конкуренции в коммерческой телекоммуникационной среде.
- 2.3 Основой глобальной связности является уровень межсетевого взаимодействия (inter-networking). Ключом к использованию этого уровня на разном оборудовании, обеспечивающем глобальную связность, является «сквозной аргумент» (end to end argument).
- 2.4 Обычно считается, что в идеальной ситуации следует применять на уровне Internet лишь один протокол. Это обеспечивает единообразие и относительно «бесшовные» операции с конкурентной сетью общего пользования с множеством провайдеров и оборудованием разных производителей. Конечно, можно применять несколько

протоколов для выполнения различных требований и есть много примеров больших частных сетей, использующих несколько протоколов сетевого уровня.

- 2.5 На практике есть по меньшей мере две причины, по которым в общедоступной сети Internet может потребоваться более 1 протокола сетевого уровня. Во-первых, это может быть связано с постепенным переходом от одной версии IP к другой. Во-вторых, могут возникнуть новые требования, ведущие к фундаментальной смене протокола.
- 2.6 Протокол уровня Internet должен быть независим от аппаратной среды и физической адресации. Такой подход позволяет применять в Internet любые новые технологии цифровой передачи и отвязать механизмы адресации от оборудования. Это открывает для Internet простые пути соединения фундаментально различающихся аппаратных сред и применения одной платформы для разных приложений и служб обмена информацией. Хорошее изложение этой модели и других фундаментальных вопросов приведено в работе [Clark].
- 2.7 Обычно считается, что сквозные (end-to-end) функции проще реализовать с помощью сквозных протоколов.
- 2.8 «Сквозной аргумент» подробно рассмотрен в работе [Saltzer]. Основной аргумент заключается в первом принципе, гласящем, что некоторые сквозные функции могут быть корректно выполнены лишь самими конечными системами. Любая сеть, сколь бы тщательно она ни была спроектирована, подвержена отказам при передаче, происходящим с некоторой статистически определённой частотой. Лучшим способом преодолеть это является передача ответственности за целостность коммуникаций конечным системам. Другим примером является сквозная защита.
- 2.9 В [Saltzer] сказано: «Рассматриваемая функция может быть полно и корректно реализована лишь при наличии знаний и помощи приложения, размещённого на конечных точках коммуникационной системы. Поэтому предоставление этой функции, как свойства самой коммуникационной системы, невозможно. Иногда предоставляемая коммуникационной системой неполная версия функции может быть полезна для повышения производительности.
- 2.10 Этот принцип имеет важные последствия, если требуется устойчивость приложений к частичным отказам в сети. Сквозному протоколу не следует полагаться на поддержку состояния (т. е. сведений о состоянии сквозных коммуникаций) внутри сети. Такие состояния следует поддерживать лишь на конечных точках, чтобы состояние разрушалось лишь самой конечной точкой (это называют общей судьбой (fate-sharing)). Прямым следствием этого является то, что передача дейтаграмм будет лучше классических виртуальных каналов (устройств). Задача сети состоит в максимально эффективной и гибкой передаче дейтаграмм.
- 2.11 Все остальное следует делать на периферии.
- 2.12 Для реализации своих служб сеть поддерживает некие сведения о состояниях: маршруты, обеспечиваемые гарантии QoS, информация о сессиях, используемая при сжатии заголовков, история сжатия данных и т. п. Эти состояния должны быть самовосстанавливающимися (self-healing) и нужны адаптивные процедуры или протоколы для создания и поддержки состояний, а также их изменения при смене топологии или активности сети. Объем этих состояний должен быть минимальным, а потеря состояния не должна приводить к чему-либо сверх временного прерывания обслуживания при условии наличия связности. Настраиваемые вручную состояния должны быть абсолютно минимальными.
- 2.13 К счастью, сеть Internet не имеет одного владельца, в ней нет централизованного управления и никто не может выключить сеть. Развитие сети зависит от того или иного согласия в части технических предложений и применяемого программного кода. Технические отклики для фактических реализаций гораздо важнее любых архитектурных принципов.

3. Общие проблемы проектирования

- 3.1 Неоднородность неизбежна и должна поддерживаться. Должны быть разрешены разные типы оборудования, например, различающиеся по скорости передачи на 7 порядков, с разным размером слов данных, разнообразные хосты - от простых микропроцессоров с малым объёмом памяти до суперкомпьютеров с параллельными вычислениями. Должны разрешаться различные протоколы приложений от простейших (например, удалённый вход в систему) до более сложных, таких как распределённые базы данных.
- 3.2 При наличии нескольких способов сделать что-либо следует выбирать один. Если имеющаяся разработка в контексте Internet или иной ситуации успешно решала такую же задачу, следует применять такое же решение, если нет веских причин отказа от него. По возможности следует избегать дублирования функциональности протоколов, но не следует считать дублирование препятствием для внесения улучшений.
- 3.3 Все проекты должны поддерживать расширение до очень большого числа устройств на сайте и многих миллионов сайтов.
- 3.4 Следует принимать во внимание производительность и стоимость, а также функциональность.
- 3.5 Следует сохранять простоту и при возникновении сомнений выбирать самое простое решение.
- 3.6 Модульность является хорошим свойством. Если можно что-то хранить отдельно, следует делать это.
- 3.7 Во многих случаях лучше принять почти полное решение сейчас, чем ждать идеального в будущем.
- 3.8 По возможности следует избегать опций и параметров. Все имеющиеся параметры и опции следует настраивать и согласовывать динамически, а не задавать вручную.
- 3.9 Следует соблюдать строгость при отправке и мягкость при восприятии. Реализации должны точно следовать спецификациям при отправке в сеть и допускать восприятие ошибочного ввода из сети. При наличии сомнений следует просто удалять ошибочный ввод без отправки сообщения об ошибке, если спецификация не требует.
- 3.10 Не следует передавать излишнее число незапрошенных пакетов, особенно групповых и широковещательных.

- 3.11 Необходимо избегать циклических зависимостей. Например, маршрутизация не должна зависеть от поиска в DNS (Domain Name System), поскольку обновление серверов DNS зависит от успеха маршрутизации.
- 3.12 Объектам следует описывать себя (включая тип и размер) в разумных пределах. Применять можно лишь коды типов и иные значения (magic number), выделенные Internet Assigned Numbers Authority (IANA).
- 3.13 Во всех спецификациях следует применять единую терминологию и обозначения, а также одни соглашения для порядка битов и байтов.
- 3.14 Ничего не следует стандартизовать, пока нет множества экземпляров работающего кода (очень важно).

4. Проблемы имён и адресов

- 4.1. Следует избегать решений с жёстким заданием адресов (hard code) или их хранением в энергонезависимой памяти (за исключением случаев, когда это требуется, например, на сервере имён или конфигурации). В общем случае пользовательским приложениям следует работать с именами, а не адресами.
- 4.2. Следует использовать единую структуру именования.
- 4.3. Общедоступные (т. е. широко видимые) имена следует задавать в кодировке ASCII без учёта регистра символов. В частности, это относится к именам DNS и элементам протоколов, передаваемым в текстовом формате.
- 4.4. Адреса должны быть однозначными (уникальными для области, где адрес может появляться).
- 4.5. Протоколы верхних уровней должны быть способны однозначно идентифицировать конечные точки. В современной практике это означает неизменность адреса начале и в конце передачи.

5. Сторонние проблемы

- 5.1 Предпочтительно выбирать технологии без патентов, но если лучшая технология запатентована и доступна на разумных условиях, приемлем выбор такой технологии.
- 5.2 Наличие экспортного контроля над некоторыми аспектами технологии Internet имеет второстепенное значение при выборе технологии для стандартизации. Технологии, требуемые для реализации стандартов Internet, могут применяться в любой стране, поэтому планетарное развёртывание технологии Internet не зависит от возможности экспорта из какой-либо страны или группы стран.
- 5.3 Ни одна реализация, не включающая все требуемые компоненты, не может претендовать на соответствие стандартам.
- 5.4 Разработкам следует быть универсальными по языку с поддержкой локализации (адаптация к национальным языкам и кодировкам). В частности, следует обеспечивать унифицированный подход к набору символов для информационного содержимого.

6. Конфиденциальность и проверка подлинности

- 6.1 Все разработки должны соответствовать архитектуре безопасности IP.
- 6.2 Операторам Internet крайне желательно обеспечивать приватность и подлинность трафика, но это не является требованием архитектуры. За конфиденциальность и проверку подлинности отвечают конечные пользователи и это должно реализовываться в применяемых ими протоколах. Конечным точкам следует быть независимыми от защиты целостности и конфиденциальности оператором. Операторы могут предоставлять ту или иную защиту, но это вторично по отношению к обеспечиваемой самими конечными пользователями защите.
- 6.3 При использовании протоколом криптографических алгоритмов протокол следует разрабатывать с возможностью применения разных алгоритмов, а выбранный в конкретной реализации алгоритм следует указывать явно. Официальные обозначения для алгоритмов записываются в реестры IANA.
(Можно утверждать, что этот принцип может быть распространён за пределы области защиты)
- 6.4 При выборе алгоритмов следует отдавать предпочтение признанным достаточно строгими для решения поставленных задач. Среди достаточно строгих алгоритмов следует выбирать проверенные временем и обеспечивающие достаточную эффективность.
- 6.5 Для обеспечения взаимодействия между конечными точками с разными службами защиты следует задать один обязательный алгоритм (набор алгоритмов), чтобы конечные точки могли согласовать контекст защиты между реализациями. Без этого возможны ситуации, когда у реализаций просто не будет общего алгоритма и они не смогут взаимодействовать в защищённом режиме.

Благодарности

Этот документ является результатом коллективной работы сообщества Internet, опубликованным Internet Architecture Board. Отдельная благодарность Fred Baker, Noel Chiappa, Donald Eastlake, Frank Kastenholz, Neal McBurnett, Masataka Ohta, Jeff Schiller и Lansing Sloan.

Литература

Ссылки были намеренно ограничены двумя фундаментальными работами по архитектуре Internet.

- [Clark] The Design Philosophy of the DARPA Internet Protocols, D.D.Clark, Proc SIGCOMM 88, ACM CCR Vol 18, Number 4, August 1988, pages 106-114 (reprinted in ACM CCR Vol 25, Number 1, January 1995, pages 102-111).
- [Saltzer] End-To-End Arguments in System Design, J.H. Saltzer, D.P.Reed, D.D.Clark, ACM TOCS, Vol 2, Number 4, November 1984, pp 277-288.

Вопросы безопасности

Вопросы безопасности не рассматриваются в этом документе.

Адрес редактора

Brian E. Carpenter

Group Leader, Communications Systems

Computing and Networks Division

CERN

European Laboratory for Particle Physics

1211 Geneva 23, Switzerland

Phone: +41 22 767-4967

Fax: +41 22 767-7155

E-Mail: brian@dxcoms.cern.ch

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru