

6. ОБМЕН ДАННЫМИ В INTERNET

[HTTP; FTP; WebDAV; BitTorrent; облачные технологии]

Всемирная паутина (WWW- World Wide Web) состоит из множества Web-серверов, файловых серверов и иных хостов, которые подключены к Internet по всему миру¹.

В рамках предлагаемой темы кратко рассматриваются основные особенности наиболее популярных и востребованных сетевых сервисов, обеспечивающих передачу данных запрашивающему их пользователю.

6.1. Передача гипертекста по протоколу HTTP

Протокол передачи гипертекста (HTTP – HyperText Transfer Protocol) относится к прикладным протоколам седьмого уровня модели OSI. Он подробно описан в документе RFC 2616. Основан на архитектуре «клиент-сервер»; обмен данными происходит по схеме «запрос-ответ».

В протоколе HTTP определено несколько команд и ответов. Наиболее часто используется запрос GET. Он используется с указанием имени файла, который нужно получить с web-сервера. Если файл найден и сервером принято решение отправить его, web-сервер отправляет код-ответ на запрос (в этом случае код 200) и непосредственно содержимое самого файла.

Web-страницы обычно состоят из множества объектов, которые хранятся на web-сервере как отдельные файлы. Для их получения web-браузер сначала получает первый файл, в котором, как правило, содержатся URL на другие объекты, по которым далее web-браузер может отправить запрос на их получение. Ниже (рис. 6.1.) схематично изображен описанный процесс.

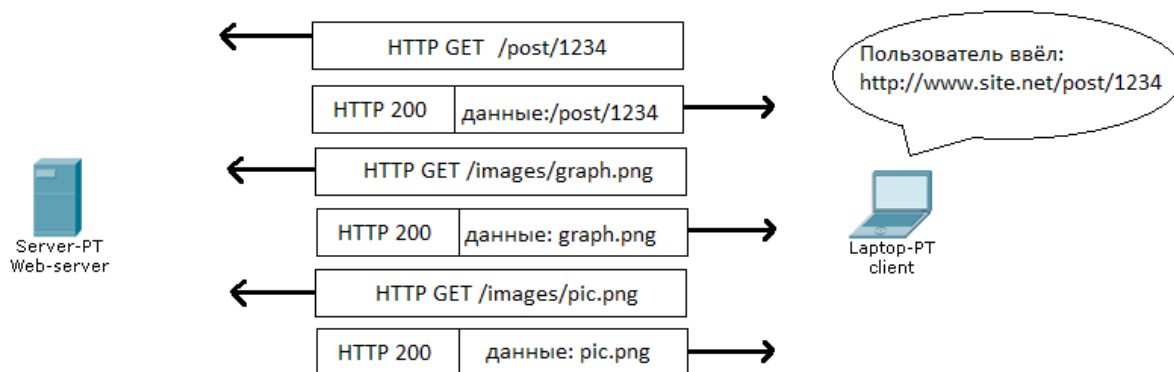


Рис.6.1. Получение файлов с Веб-страницы по протоколу HTTP

6.2. WebDAV

¹ Подробнее см. Тему 3 настоящего конспекта лекций по дисциплине «Мировые информационные ресурсы».

Говоря о HTTP, нельзя не упомянуть протокол *WebDAV* (*Web-based Distributed Authoring and Versioning* – *web-распределенные авторизация и контроль версий*). Этот защищенный протокол работает поверх HTTP и служит для доступа к объектам и коллекциям, а также расширяет HTTP набором дополнительных команд.

Используется для выполнения основных и расширенных (контроль версий) файловых операций над объектами на серверах; работы с любым типов объектов (не только файлы); также поддерживает одновременную работу над несколькими объектами.

Например, в среде Windows можно открыть ресурс WebDAV как обычную папку. Если это сделают несколько пользователей, они получают общую папку, содержимое которой хранится на сервере. Документы можно открывать прямо из этой папки.

WebDAV позволяет удаленным пользователям перемещать, искать, редактировать или удалять файлы и каталоги с сервера. Для того чтобы вносить изменения в файлы с помощью WebDAV, пользователи должны пройти авторизацию, то есть получить специальное разрешение на внесение изменений.

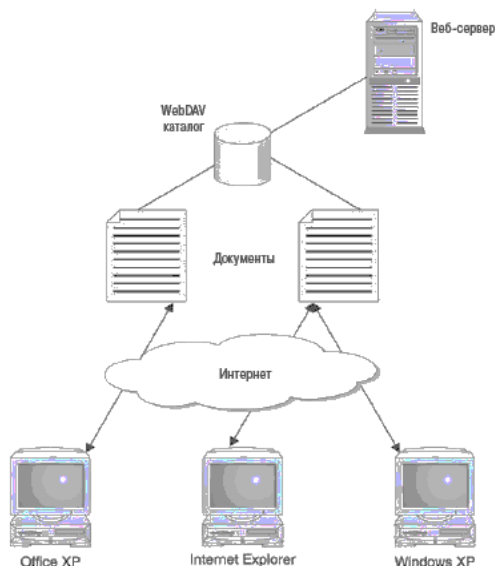


Рис.6.2. Общая схема использования протокола *WebDAV*

Клиенты, использующие WebDAV, могут получать доступ к каталогам посредством любого клиентского приложения, поддерживающего этот протокол.

Весь перечисленный функционал позволяет использовать WebDav в качестве замены *FTP* (см. следующий раздел).

WebDav сейчас можно встретить при пользовании каким-либо облачным хранилищем. Например, доступ к сервису *Яндекс.Диск* можно получить, используя рассматриваемый протокол. Клиентом могут выступать как специальные приложения, так и стандартные проводники операционных систем (Проводник Windows; Finder в OS X; Dolphin или Nautilus в Linux).

6.3. Передача файлов по протоколу FTP

Стоит отметить, что получить доступ к файлам можно не только через службу WWW и, соответственно, протокол HTTP. Существует и множество других протоколов передачи данных.

Протокол передачи файлов (FTP – File Transfer Protocol), один из старейших протоколов, разработанный в 1971 году (к слову, HTTP разработан только в 1992 году). С течением времени FTP несколько изменялся: например, в 1998 году добавлена поддержка протокола передачи IPv6. Спецификация FTP подробно описана в RFC 959.

FTP – «клиент-серверный» протокол с установлением соединения, в котором обмен данными между клиентом и сервером осуществляется по TCP-соединению. Для FTP зарезервировано два порта TCP: 20 – для передачи данных и 21 для служебных команд (ниже рассмотрено, почему используются эти два порта).

Реализация передачи файла по FTP предусматривает разделение функций между двумя элементами:

- *Интерпретатором протоколов (PI - Protocol Interpreter)*
- *Процессом передачи данных (DTP - Data Transfer Process)*

При передаче файлов PI и DTP существуют параллельно, как два взаимодействующих процесса. Именно поэтому протокол FTP требует установления двух соединений между клиентом и сервером.

PI выполняет функции основного управляющего соединения между клиентом и сервером. Он устанавливает TCP соединение и передает управляющую информацию серверу. Управляющая информация представляет собой различные команды, например: изменение имени файла, перемещение по иерархии каталогов и т.д. Управляющее соединение остается открытым до закрытия его пользователем.

Процедура подключения пользователя к серверу состоит из пяти этапов:

1. *Пользовательский PI отправляет запрос соединения серверному PI на 21-й порт.*
2. *Серверный PI возвращает отклик, и соединение считается установленным.*
3. *После открытия управляющего TCP соединения серверный PI начинает выполнять последовательность входа в систему.*
4. *Пользователь проходит аутентификацию.*
5. *Начинается процесс передачи данных.*

DTP приводится в действие только в том случае, если пользователь в явном виде запросил передачу файлов с сервера или на сервер. В отличие от PI, который открыт постоянно, соединение DTP автоматически закрывается по завершении передачи файла.

Схематично описанный процесс изображен на рис. 3.2

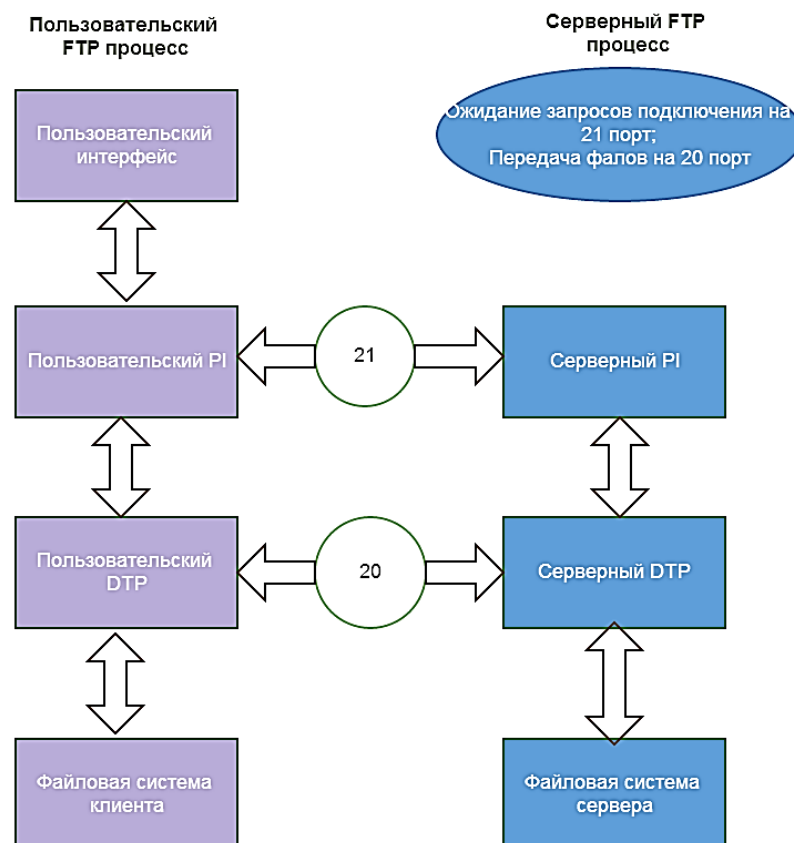


Рис.6.3. Передача данных по протоколу FTP

6.4. Сету BitTorrent

Популярный *P2P* протокол, созданный в 2001 году программистом Брэмом Коэном, предназначен для совместного обмена файлами через Internet.

Как правило, при использовании *BitTorrent* присутствует специальный сервер – трекер², который не задействован в непосредственном процессе передачи данных, а выполняет роль хранилища (*repository*, англ.), содержащего различную служебную информацию (IP-адреса узлов, входящие порты клиентов и др.).

Каждый узел – *пир* (*peer* – равный, англ.) играет роли и клиента, и сервера. Такая *одноранговая* (иначе – *пиринговая*) сетевая архитектура позволяет сохранять работоспособность сети при любом количестве и любом сочетании доступных узлов.

² Трекер (*tracker*, англ.) — специализированный сервер, работающий по протоколу HTTP, используемый для того, чтобы клиенты могли найти друг друга. Фактически, на трекере хранятся IP-адреса, входящие порты клиентов и хэши-суммы, уникальным образом идентифицирующие объекты, участвующие в заках. По стандарту, имена файлов на трекере не хранятся, и узнать их по хэши-суммам нельзя. Однако на практике трекер часто помимо своей основной функции выполняет и функцию небольшого web-сервера. Такой сервер хранит файлы метаданных и описания распространяемых файлов, предоставляет статистику зачек по разным файлам, показывает текущее количество подключенных пиров и пр.

В процессе обмена данные передаются небольшими фрагментами – сегментами. Каждый клиент одновременно и принимает сегменты и отдает другим клиентам. Для того чтобы вступить в процесс обмена данными, клиенту необходимо подключиться к трекеру (его адрес содержится в торрент-файле), указать свой адрес и хеш-сумму торрент-файла. Только после этого клиент сможет получить сведения о других участниках обмена и присоединиться к ним. Клиент также должен информировать трекер о ходе процесса и получать обновленный список адресов участников обмена.

При соединении клиенты сразу обмениваются информацией об имеющихся у них сегментах. Клиент, желающий скачать сегмент, посылает запрос и, если второй клиент готов отдавать, получает этот сегмент. После этого клиент проверяет контрольную сумму сегмента. Если она совпала с той, что записана в торрент-файле, то сегмент считается успешно скачанным, и клиент оповещает всех присоединенных клиентов о наличии у него этого сегмента. Если же контрольные суммы различаются, то сегмент начинает скачиваться заново.

Стоит отметить, что трекер является уязвимым местом всей системы, поскольку при его отказе клиенты не смогут получить информацию и подключиться друг к другу. Поэтому в новых версиях протокола реализована работа без трекера. При такой организации фактически каждый клиент и является трекером (информация трекера присутствует на клиентах в виде распределенной хэш-таблицы (*DHT - Distributed Hash Table*)).

6.5. Облачные технологии обмена данными

Облачные технологии, на текущий момент, весьма популярное направление в сфере ИТ. Концепцию облачных технологий можно сформулировать как предоставление пользователям удаленного доступа к услугам, вычислительным мощностям и приложениям через Internet. Если сказать обобщенно, то облака превращают информационные технологии в услугу.

Облака активно применяются в самых разных сферах, в том числе, и в бизнесе, когда предприятие переносит в облако часть своей инфраструктуры. «Переезд» в облако имеет следующие плюсы:

- *Оптимизация затрат* облака задает четкое соответствие между потребляемыми ресурсами и затратами на них. Другими словами, предприятие платит только за те ресурсы, которые оно потребляет.
- *Автоматизация*. Быстрота развертывания, свертывания и масштабирования продукта. В облаке возможно выполнять поставленные задачи за минимальный период времени. С помощью облачных технологий можно за короткое время развернуть проект, задать какое количество ресурсов ему требуется, а при необходимости добавить их, или уменьшить, или вообще полностью свернуть проект и прекратить тратить средства на его поддержку.

- *Защита бизнеса.* Облако находится в *data-центре* (*data center*, *англ.*) с гарантируемым уровнем доступности. И если облако внешнее, то данные, которые находятся в нем, не зависят от того, что происходит на стороне предприятия.

Схематично общая архитектура облака показана на рис.6.4

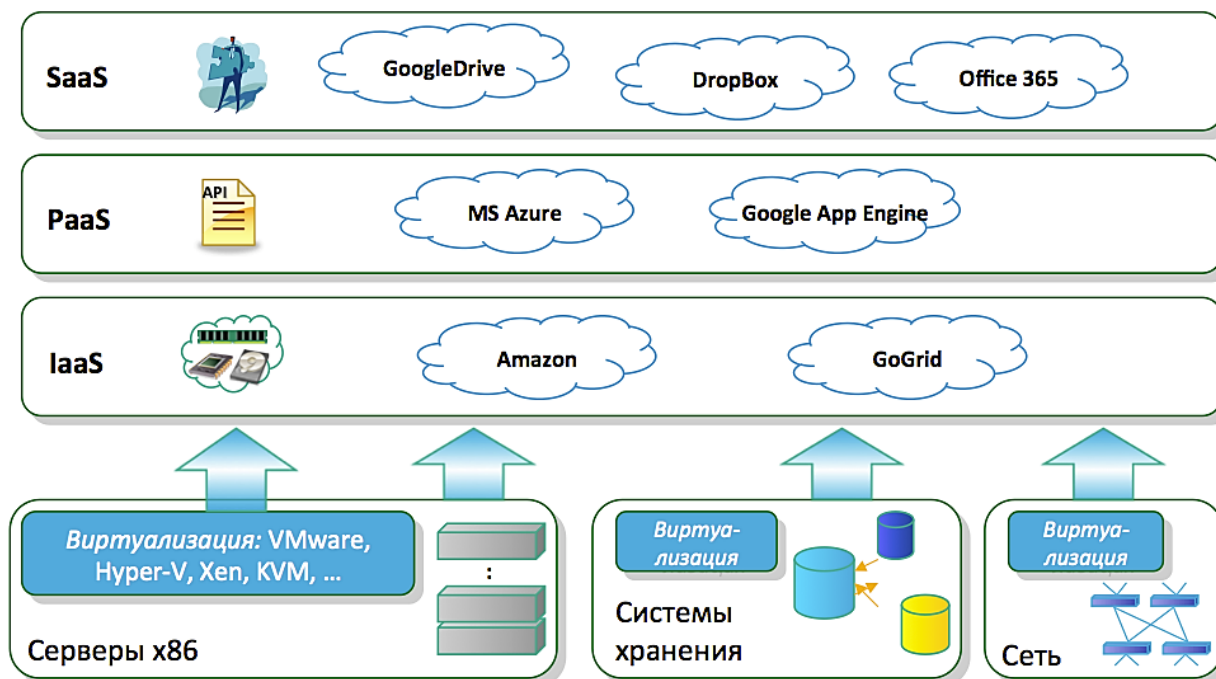


Рис.6.4. Общая архитектура облака

Облачная архитектура включает в себя следующие основные компоненты:

1. Какая-либо аппаратная платформа, например, серверы на архитектуре x86 или x64, на которых производится виртуализация с помощью различных гипервизоров (KVM, Xen, Hyper-V и т.д.).
2. Виртуализированная система хранения данных.
3. Виртуализированная сеть передачи данных.

Таким образом, все основные компоненты инфраструктуры должны быть виртуализованы. Теоретически, можно организовать всю инфраструктуру без применения виртуализации, но это экономически очень невыгодное и плохо управляемое решение.

Когда заканчивается организация инфраструктуры, на ее основе можно организовать инфраструктурное облако, которое можно предоставить как сервис. Инфраструктура как сервис (*Infrastructure-as-a-Service – IaaS*, *англ.*) предоставляет пользователю возможность самостоятельно управлять ресурсами облачной инфраструктуры: такими как системы обработки и хранения, системы управления сетями и другими вычислительными ресурсами. Пользователь может устанавливать и запускать произвольное программное обеспечение (операционные системы, прикладное программное обеспечение) и полностью контролировать его. Пример популярной IaaS – *Amazon web services*.

Когда у IaaS появляется программный интерфейс управления, с помощью которого можно выделить какие-либо ресурсы, необходимые для функционирования прикладных программ, IaaS превращается в некое подобие операционной системы и становится платформой, которая предоставляется как услуга. Платформа как услуга (*Platform-as-a-Service – PaaS, англ.*) дает потребителю возможность использовать IaaS для размещения базового программного обеспечения, для последующего размещения на нем новых или существующих приложений (собственных, разработанных на заказ или приобретенных). В состав таких платформ входят инструментальные средства создания, тестирования и выполнения прикладного программного обеспечения — системы управления базами данных, связующее программное обеспечение, среды исполнения языков программирования и т.д. Примеры PaaS – *Google App Engine, Microsoft Azure* и др.

Наивысшая ступень облаков (та, с которой большинство обычных пользователей сталкиваются каждый день) называется «*программное обеспечение как услуга*» (*Software-as-a-Service – SaaS, англ.*). Программное обеспечение как услуга предоставляет пользователю возможность пользоваться прикладным программным обеспечением, которое находится в инфраструктуре, посредством различных клиентских устройств или тонкого клиента. Примеры SaaS – *Google Drive, Office 365*, сервисы электронной почты и т.д. В общих чертах, SaaS – программные пакеты, которые умеют управлять частью IaaS (например, количеством ресурсов) через своеобразный *API (application programming interface – прикладной программный интерфейс, англ.)*, которым является PaaS.