СиТ. Лаба 2.

SMTP

В качестве средств передачи сообщения почтовая служба Интернета использует стандартный,
разработанный специально для почтовых систем протокол SMTP (Simple Mail Transfer
Protocol — простой протокол передачи почты). SMTP реализуется несимметричными взаимодействующими частями: SMTP-клиентом, работающим на стороне
отправителя, и SMTP-сервером, работающим на стороне получателя. SMTP-сервер должен
постоянно быть в режиме подключения, ожидая запросов со стороны SMTP-клиента.
Логика протокола SMTP является достаточно простой:

1. После того как, применяя графический интерфейс своего почтового клиента, пользователь щелкает на значке отправки сообщения, SMTP-клиент посылает запрос на
   установление TCP-соединения на порт 25 SMTP-сервера (это назначенный порт).
2. Если сервер готов, то он посылает свои идентификационные данные, в частности свое
   DNS-имя. Если SMTP-сервер оказался не готов, то он посылает соответствующее сообщение клиенту, и тот снова посылает запрос, пытаясь заново установить соединение.
   Затем клиент передает серверу почтовые адреса (имена) отправителя и получателя.
   Если имя получателя соответствует ожидаемому, то после получения адресов сервер
   дает согласие на установление SMTP-соединения, и в рамках этого логического канала
   происходит передача сообщения.

3. Если после приема тела сообщения сервер отвечает командой OK, это означает, что
   сервер принял на себя ответственность по дальнейшей передаче сообщения получателю.
   Однако это не означает, что сервер гарантирует успешную доставку, потому что
   последнее зависит не только от него: например, клиентская машина получателя может
   быть в течение длительного времени не подсоединена к Интернету. Если сервер не
   может доставить сообщение, то он передает отчет об ошибке отправителю сообщения
   и разрывает соединение.

Команды SMTP

HELO <HOST> — Эта команда используется для идентификации машины отправителя (HOST) на SMTP сервере.

MAIL <SP> FROM: <reverse-path> <CRLF> — Эта команда указывает SMTP-серверу начать новую транзакцию по приёму почты. В качестве аргумента, она передаёт на сервер почтовый адрес отправителя письма. Если адрес отправителя правильный и не содержит ошибок, то сервер вернёт ответ «250 OK».

RCPT <SP> TO: <forward-path> <CRLF> — Эта команда передаёт на сервер почтовый адрес получателя письма. Если адрес получателя не содержит ошибок, то тогда SMTP сервер вернёт ответ «250 OK». Если в адресе получателя есть ошибка, то сервер вернёт сообщение с кодом 550. Данная команда может повторяться сколь угодно долго по числу получателей, однако современные почтовые сервера вводят ограничения на количество одновременных получателей.
Пример: RCPT TO: dog@switzerland.mail.sz

DATA <CRLF> — Если она принимается сервером, то он возвращает сообщение с кодом 354, приглашающее продолжить отправку сообщения. После этого, на сервер можно передавать текст почтового сообщения. Признаком окончания передачи почтового сообщения является символ точки «.» в начале новой строки.

Если сообщение принято к доставке, то сервер вернёт уведомление с кодом 250, а иначе – сообщение об ошибке.

QUIT — Получив эту команду сервер должен возвратить отклик и 221 OK закрыть канал передачи.

AUTH LOGIN — Сообщаем серверу о намерении пройти авторизацию. После этого отправляем логин и пароль закодированные в base64.

POP3 и IMAP

POP3 (Post Office Protocol v.3 — протокол почтового отделения версии 3) и IMAP (Internet Mail Access Protocol — протокол доступа к электронной почте Интернета). Оба протокола решают одну и ту же задачу — обеспечивают пользователей доступом к их корреспонденции, хранящейся на почтовом сервере.
В связи с многопользовательским характером работы почтового сервера оба протокола поддерживают аутентификацию пользователей на основе идентификаторов и паролей пользователей. Однако протоколы POP3 и IMAP имеют и принципиальные различия, важнейшее из которых состоит в следующем. Получая доступ к почтовому серверу по протоколу POP3, вы «перекачиваете» адресованные вам сообщения в память своего компьютера, при этом на сервере не остается никакого следа от считанной вами почты. Если
же доступ осуществляется по протоколу IMAP, то в память вашего компьютера передаются
только копии сообщений, хранящихся на почтовом сервере.

При просмотре почты с использованием протокола POP3 все электронные письма загружаются на локальную машину пользователя и удаляются на сервере. При таком подходе использование данного протокола будет удобно только при работе на одной локальной машине. Однако следует отметить что современные почтовые клиенты предлагают возможность не удалять письма с сервера. По умолчанию протокол работает с портом 110 — для передачи данных без шифрования и с портом 995 — для передачи данных с использованием SSL/TLS методов шифрования. Также следует отметить, что протокол POP3 работает только в одном направлении. Это означает, что данные с сервера могут быть загружены на ваш локальный клиент, но не могут быть отправлены с локального клиента на удаленный сервер.

Команды POP3

USER [имя] — Отправляет имя пользователя.

PASS [пароль] — Отправляет пароль пользователя.

DELE [сообщение] — Помечает указанное сообщение на удаление. Помеченные сообщения, на самом деле, удаляются только после закрытия транзакции, после того, как прошла команда QUIT.

LIST [сообщение] — Если передать аргумент - номер сообщения, то сервер выдаёт инфу о запрашиваемом сообщении. Иначе, сервер выдаёт информацию про все сообщения, которые есть на почте.

RETR [сообщение] — Получить сообщение с указанным номером.

RSET — Откат транзакций в пределах текущего сеанса. К примеру, если вы случайно пометили сообщение на удаление, то можно убрать метки, послав команду RSET.

STAT — Получить количество сообщений в ящике, а так же ещё и размер, занимаемый этими сообщениями на сервере.

TOP — Получить заголовки указанного сообщения, и указанное количество первых строк сообщения. Эти данные вернутся, разделённые пустой строкой.

QUIT — Завершает транзакцию.

Система DNS

DNS (англ. Domain Name System «система доменных имён») — компьютерная распределённая система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты и/или обслуживающих узлах для протоколов в домене (SRV-запись).

Доменная служба имен (Domain Name Service, DNS) отображает символьные имена узлов сети на их IP-адреса (как IPv4, так и IPv6).

Домен — узел в дереве имён, вместе со всеми подчинёнными ему узлами (если таковые имеются), то есть именованная ветвь или поддерево в дереве имён. Структура доменного имени отражает порядок следования узлов в иерархии; доменное имя читается слева направо от младших доменов к доменам высшего

уровня (в порядке повышения значимости): вверху находится корневой домен (имеющий идентификатор «.»(точка)), ниже идут домены первого уровня (доменные зоны), затем — домены второго уровня, третьего и т. д. (например, для адреса ru.wikipedia.org. домен первого уровня — org, второго — wikipedia, третьего — ru). DNS позволяет не указывать точку корневого домена.

Поддомен (англ. subdomain) — подчинённый домен (например, wikipedia.org — поддомен домена org, а ru.wikipedia.org — домена wikipedia.org).

Краткое доменное имя — это имя конечного узла сети: хоста или порта маршрутизатора. Краткое имя — это лист дерева имен. Относительное доменное имя — это составное имя, начинающееся с некоторого уровня иерархии, но не самого верхнего. Например, www.zil — это относительное имя. Полное доменное имя включает составляющие всех уровней иерархии, начиная от краткого имени и кончая корневой точкой: www.zil.mmt.ru..

Система DNS состоит из серверов и клиентов. ***DNS-серверы*** поддерживают распределенную базу отображений, а ***DNS-клиенты*** обращаются к серверам с запросами об отображении доменного имени на IP-адрес (эту процедуру называют также «разрешением» доменного имени).

DNS-серверы образуют ***иерархию***. На самой вершине иерархии располагаются корневые
серверы. Серверы верхнего уровня хранят данные об именах и адресах имен, входящих в домены верхнего уровня (com, ru или fm), а также об именах серверов DNS, которые обслуживают домены второго уровня иерархии (cisco.com, yandex.ru и т. п.).

Локальный сервер – это одна из разновидностей сервисов, которая сообщается с остальными DNS-серверами региона. Такой сервер выполняет запросы несколько раз, после чего получает необходимую информацию и передает ее в браузер.

Записи ресурсов доменов. У каждого домена, независимо от того, является ли он одиноким хостом или доменом верхнего уровня, может быть набор ассоциированных с ним записей ресурсов (resource records). Эти записи являются базой данных DNS.

Запись A (address record) или запись адреса связывает имя хоста с адресом протокола IPv4. Например, запрос A-записи на имя referrals.icann.org вернёт его IPv4-адрес — 192.0.34.164.

Запись ***MX*** (от англ. mail exchanger) — тип DNS-записи, предназначенный для маршрутизации электронной почты с использованием протокола SMTP. Она указывает, какими серверами обрабатывается почта для нашего домена.

В каждой MX-записи для конкретного доменного имени (например: *@wikipedia.org*) хранятся два поля:

- имя почтового сервера, обслуживающего домен (например: *mx1001.wikimedia.org*). По данному имени с помощью A-записи будет определяться IP-адрес, поэтому A-запись должна также существовать (недопустимо вместо A использовать CNAME, так как в этом случае возможно зацикливание, подробнее описано в разделе «Особенности»);
- порядковый номер предпочтения этого сервера (например: *10*) — используется в случаях, когда доменное имя обслуживается несколькими почтовыми серверами.

Пример записи:

> example.com. IN MX 10 mail.example.com
>
> Где:
> example.com — домен, для которого обрабатывается почта.
> IN MX — тип записи.
> 10 — приоритет записи (Подробнее — ниже).
> mail.example.com — A-имя почтового сервера.

Как применяется mx dns запрос при отправке почты? Чтобы отправить электронную почту, сервер-отправитель запрашивает у DNS-сервера MX-запись домена получателя электронного сообщения (то есть части адреса после символа «@»). В результате запроса возвращается список имён хостов почтовых серверов, принимающих входящую почту для данного домена, и номеров предпочтения для каждого из них. Сервер-отправитель затем пытается установить SMTP-соединение с одним из этих хостов, выбирая имена по порядку, начиная с наименьшего номера предпочтения, и перебирая их до тех пор, пока не удастся установить соединение. Если имеется несколько хостов с одинаковым предпочтением, то должны быть предприняты попытки установить соединение с каждым из них.

Если ни с одним сервером не удалось установить соединение, сервер-отправитель будет продолжать попытки в течение некоторого времени (в

зависимости от настроек, обычно от нескольких часов до двух недель), после чего сформирует и отправит отправителю письма отчёт об ошибке.

Механизм записей MX предоставляет возможность использовать множество серверов для одного домена и упорядочивания их использования в целях уменьшения нагрузки и увеличения вероятности успешной доставки почты. Кроме того, такой механизм предоставляет возможность распределить обработку входящей почты среди нескольких физических серверов.

Какие еще DNS запросы бывают? Виды записей:

- **Запись AAAA** (*IPv6 address record*) связывает имя хоста с адресом протокола IPv6. Например, запрос AAAA-записи на имя `K.ROOT-SERVERS.NET` вернёт его IPv6адрес — `2001:7fd::1`.
- **Запись CNAME** (*canonical name record*) или **каноническая запись имени** (псевдоним) используется для перенаправления на другое имя.
- **Запись NS** (*name server*) указывает на DNS-сервер для данного домена.
- **Запись PTR** (*pointer*) обратная DNS-**запись** или **запись указателя** связывает IP-адрес хоста с его каноническим именем. Запрос в домене `in-addr.arpa` на IP-адрес хоста в reverse-форме вернёт имя (FQDN) данного хоста (см. Обратный DNS-запрос). Например (на момент написания), для IP-адреса `192.0.34.164` запрос записи PTR `164.34.0.192.in-addr.arpa` вернёт его каноническое имя `referrals.icann.org`. В целях уменьшения объёма нежелательной корреспонденции (спама) многие серверы-получатели электронной почты могут проверять наличие PTR-записи для хоста, с которого происходит отправка. В этом случае PTR-запись для IP-адреса должна соответствовать имени отправляющего почтового сервера, которым он представляется в процессе SMTPсессии.
    - **Запись SOA** (*Start of Authority*) или **начальная запись зоны** указывает, на каком сервере хранится эталонная информация о данном домене, содержит контактную информацию лица, ответственного за данную зону, *тайминги* (параметры времени) кеширования зонной информации и взаимодействия DNS-серверов.
- SRV-запись (*server selection*) указывает на серверы для сервисов, используется, в частности, для Jabber и Active Directory.

Существуют две основные схемы разрешения DNS-имен. В первом варианте, называемом
итеративной процедурой, работу по поиску IP-адреса координирует DNS-клиент. он итеративно выполняет последовательность запросов к разным серверам имен.

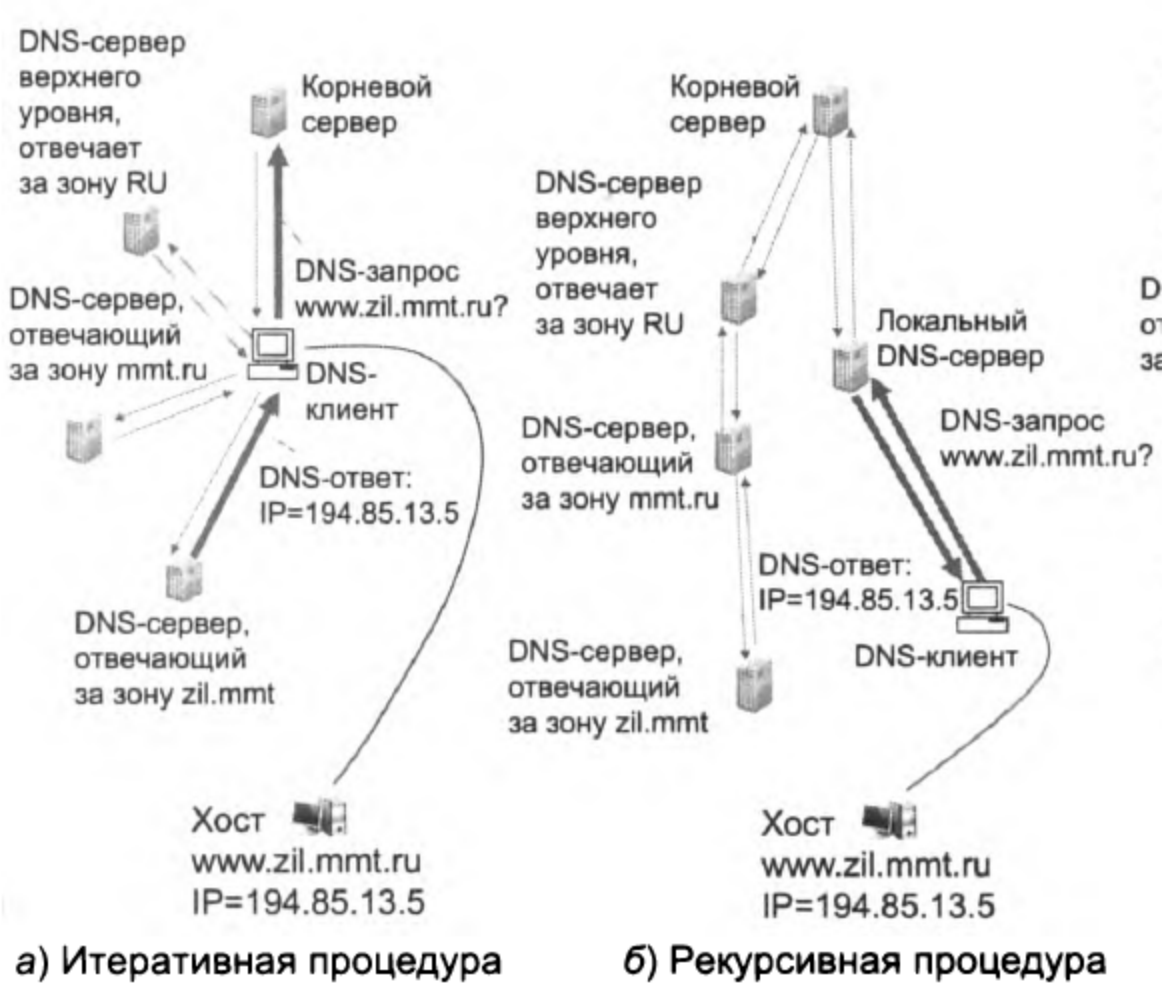
1. DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного
   имени www.zil.mmt.ru хоста, для которого он хочет найти 1P-адрес.
2. Корневой DNS-сервер отвечает клиенту, указывая адреса DNS-серверов верхнего
   уровня, обслуживающих домен, заданный в старшей части запрошенного имени,
   в данном случае — домен ru.
3. DNS-клиент делает следующий запрос к одному из предложенных ему DNS-серверов
   верхнего уровня, который отсылает его к DNS-серверу нужного поддомена (в примере
   это сервер, отвечающий за зону mmt.ru), и так далее, пока не будет найден DNS-сервер,
   в котором хранится отображение запрошенного имени на IP-адрес. Этот сервер дает
   окончательный ответ клиенту, который теперь может установить связь с хостом по
   IP-адресу 194.85.13.5.

Во втором варианте выполняется ***рекурсивная процедура***. Здесь DNS-клиент перепоручает всю работу по разрешению имени цепочке DNS-серверов.

1. DNS-клиент отправляет запрос к локальному DNS-серверу, то есть серверу, обслуживающему поддомен, которому принадлежит имя клиента.
2. Если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту. Это
   может быть полномочный ответ (запрошенное имя входит в тот же поддомен, что и имя
   клиента) или неполномочный ответ (сервер уже узнавал данное соответствие

для другого
клиента и сохранил его в своем кэше).

3. Если локальный DNS-сервер не знает ответа, то он обращается к корневому серверу,
который переправляет запрос к DNS-серверу верхнего уровня (отвечающему за зону
RU), который в свою очередь запрашивает нижележащий сервер (зона mmt), и так
далее, пока запрос не дойдет до полномочного сервера, имеющего в своем файле зоны
запись о запрошенном имени.
4. DNS-ответ полномочного сервера проходит тот же путь по цепочке DNS-серверов
в обратном направлении, пока не достигнет DNS-клиента, породившего данный запрос.



Команда nslookup

nslookup — сетевая утилита командной строки, используется для запросов в систему доменных имён (DNS) с целью выявления имен или IP-адресов, а также других специфических DNS записей.

Для определения MX-записей хоста используется команда:

nslookup –type=MX ДОМЕННОЕ-ИМЯ-ХОСТА

MIME

MIME ("Multipurpose Internet Mail Extensions" [Многоцелевые расширения почтового стандарта Internet]) — стандарт, описывающий как пересылать по электронной почте исполняемые, графические, мультимедийные, смешанные данные. MIME

также позволяет размечать письмо на части различных типов так, чтобы получатель (почтовая программа) мог определить, что делать с каждой из частей письма.

Все почтовые сообщения, составленные в соответствии с MIME-стандартом, должны иметь ***поле MIME-Version*** в своем заголовке.

Content-Type. Назначение этого поля - наиболее полное описание данных, содержащихся в теле, с тем, чтобы почтовый агент (программа) получателя могла выбрать соответствующий механизм для их обработки.

Выделяются следующие базовые типы передаваемых данных:

application; audio; image; message (письмо в письме); model; multipart; text; video.

multipart -- содержимое письма состоит из некоторого множества частей, содержащих данные различных взаимонезависимых типов. Изначально определено четыре подтипа:

1. **"mixed"** - основной;
2. **"alternative"** - для представления одних и тех же данных в разных форматах;
3. **"parallel"** - если разные части документа должны просматриваться одновременно;
4. **"digest"** - если каждая из частей тела письма имеет тип "message".

Для передачи множественного сообщения в заголовок Content-Type добавляется параметр ***boundary*** (граница), который обозначает последовательность символов, разделяющих части сообщения. Граница может состоять из цифр, букв и символов '()+_,-./:=?. При использовании специальных символов (не цифр и букв) значение параметра boundary следует заключать в двойные кавычки ". Максимальная длина границы — 70 символов[1].

Начало каждой части сообщения обозначается строкой --boundary. Конец последнего сообщения обозначается строкой --boundary--.

Content-Transfer-Encoding. Многие типы данных, пересылаемых через email требуют "натурального" представления, то есть, 8-битный набор символов либо двоичный код (что для машины - одно и то же, только представимо для пользователя по-разному). В таком виде данные не могут быть пересланы по 7-

битным почтовым протоколам, например, RFC 821, который, к тому же, ограничивает длину строки 1000 символами.

Content-Disposition является индикатором того, что ожидаемый контент будет отображаться в браузере, как часть электронного письма, или же как вложение, которое затем может быть скачано и сохранено локально.

Взаимодействие почтовых агентов и серверов

Система электронной почты (система e-mail) состоит из двух подсистем: ***пользовательских агентов (user agents)***, позволяющих пользователям читать и отправлять электронную почту, и а***гентов передачи сообщений (message transfer agents)***, пересылающих сообщения от отправителя к получателю. Мы будем неформально называть агенты передачи сообщений ***почтовыми серверами (mail servers)***.

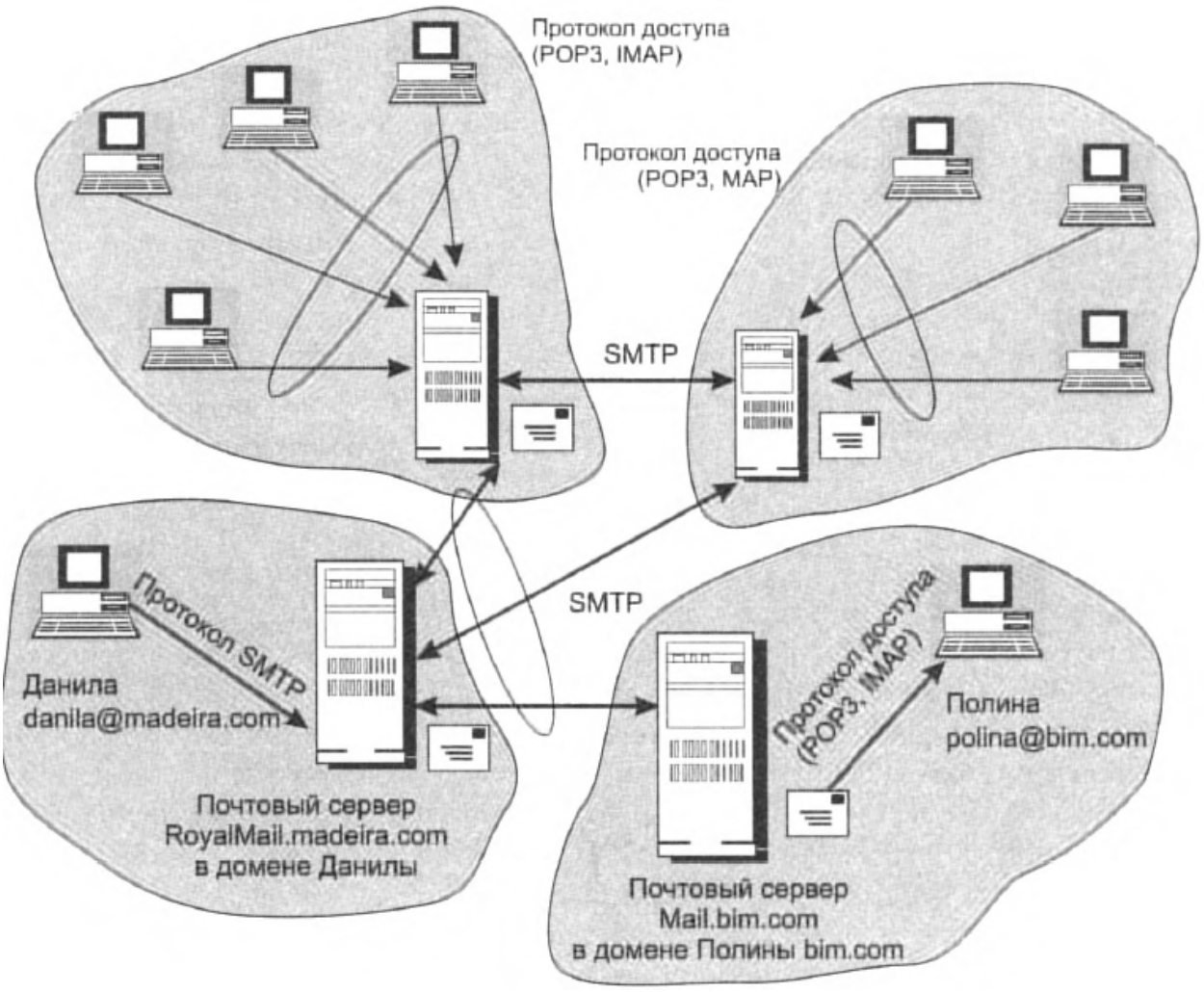
Агенты передачи сообщений, как правило, являются системными процессами. Они работают в фоновом режиме на машинах почтовых серверов и всегда должны быть доступными. Они должны автоматически перемещать почтовые сообщения по системе от
отправителя получателю при помощи SMTP. Это шаг, на котором передается сообщение. SMTP отсылает сообщения по соединениям и высылает обратно отчеты о статусе доставки и любых возникших ошибках.

Агенты передачи сообщений также используют списки рассылки (mailing lists), которые позволяют доставлять идентичные копии сообщения всем, чьи адреса были
включены в список адресов электронной почты. Среди других полезных дополнительных функций можно перечислить следующие: рассылка копий писем «под копирку»
(Carbon copy), рассылка копий без уведомления о других получателях (Blind carbon copy), письма с высоким приоритетом, секретная (то есть зашифрованная) почта, возможность доставки письма альтернативному получателю, если основной временно
недоступен, а также возможность перепоручать обработку почты секретарям.
За связь пользовательских агентов и агентов передачи сообщений отвечают почтовые ящики и стандартный формат почтовых сообщений. Почтовые ящики (mailboxes)

хранят почту, которая доставлена пользователю. Они поддерживаются почтовыми серверами. Пользовательские агенты просто предоставляют пользователям возможность увидеть содержимое их почтовых ящиков. Чтобы это сделать, пользовательский агент отсылает почтовым серверам команды и получает возможность манипулировать
почтовыми ящиками, проверяя их содержимое, удаляя сообщения и т. д. Последний
шаг в извлечении почты — это ее доставка конечному пользователю.
При такой архитектуре один пользователь может использовать различные пользовательские агенты на различных машинах, чтобы получить доступ к одному и тому же
почтовому ящику.
В основе всех современных систем электронной почты лежит ключевая идея о разграничении конверта (envelope) и содержимого письма. Конверт заключает в себе сообщение. Он содержит всю информацию, необходимую для доставки сообщения, — адрес получателя, приоритет, уровень секретности и т. п. Все эти сведения отделены от самого сообщения. Агенты передачи сообщений используют конверт для маршрутизации, аналогично тому, как это делает обычная почтовая служба.
Сообщение внутри конверта состоит из двух отдельных частей: заголовка (header) и тела письма (body). Заголовок содержит управляющую информацию для пользовательских агентов. Тело письма целиком предназначается для человека-получателя.

Взаимодействие почтовых серверов

Схема с двумя почтовыми серверами-посредниками.



Здесь передача сообщений между клиентами почты проходит через два промежуточных почтовых сервера, каждый из которых обслуживает домен своего клиента. На каждом из
этих серверов установлены и клиентские части протокола SMTP. При отправке письма почтовый клиент Данилы передает сообщение по протоколу SMTP почтовому серверу домена, к которому относится Данила, — RoyalMail.madeira.com. Это сообщение буферизуется на данном сервере, а затем по протоколу SMTP передается дальше на почтовый сервер домена Полины — mail.bim.com, откуда описанным уже образом попадает на компьютер Полины.

Возникает вопрос, зачем нужна такая двухступенчатая передача через два почтовых сервера?

Прежде всего для повышения надежности и гибкости процедуры доставки

сообщения.
Действительно, в схеме с передачей сообщения сразу на сервер получателя почтовый
клиент отправителя в случае неисправности почтового сервера должен самостоятельно
справляться со сложившейся нештатной ситуацией. Если же посредником в передаче сообщения является другой почтовый сервер, то это позволяет реализовывать разнообразные логические механизмы реакции на отказы на стороне сервера, который к тому же всегда находится в режиме подключения. Например, при невозможности передать письмо почтовому серверу получателя сервер отправляющей стороны может не только рапортовать об этом своему клиенту, но и предпринимать собственные действия — пытаться снова и снова послать письмо, повторяя эти попытки в течение достаточно длительного периода.

Round-Robin DNS

Round Robin, или алгоритм кругового обслуживания, представляет собой перебор по круговому циклу: первый запрос передаётся одному серверу, затем следующий запрос передаётся другому и так до достижения последнего сервера, а затем всё начинается сначала.

Самой распространенной реализацией этого алгоритма является метод балансировки Round Robin DNS. Как известно, любой DNS-сервер хранит пару «имя хоста — IP-адрес» для каждой машины в определённом домене.
Этот список может выглядеть, например, так:
example.com xxx.xxx.xxx.2
www.example.com xxx.xxx.xxx.3
С каждым именем из списка можно ассоциировать несколько IP-адресов:
example.com xxx.xxx.xxx.2
www.example.com xxx.xxx.xxx.3
www.example.com xxx.xxx.xxx.4
www.example.com xxx.xxx.xxx.5
www.example.com xxx.xxx.xxx.6
DNS-сервер проходит по всем записям таблицы и отдаёт на каждый новый запрос следующий IP-адрес: например, на первый запрос — xxx.xxx.xxx.2, на второй —

xxx.xxx.xxx.3, и так далее. В результате все серверы в кластере получают одинаковое количество запросов.

Достоинством алгоритма является независимость от протоколов более высоких уровней, возможность использования любого протокола, обращающегося к серверу по доменному имени. Балансировка на основе алгоритма Round Robin никак не зависит от нагрузки на сервер: кэширующие DNS-серверы помогут справиться с любым наплывом клиентов, не требуется связи между серверами и его можно использовать как для локальной, так и для глобальной балансировки, низкая стоимость, достаточно просто добавить несколько записей DNS.
Но также алгоритм Round Robin имеет и целый ряд существенных недостатков. Для эффективного и справедливого распределения нагрузки у каждого сервера должен быть одинаковый набор ресурсов, совершенно не учитывается загруженность того или иного сервера в составе кластера, не учитывается количество активных на данный момент подключений сервера.
Представим себе следующую гипотетическую ситуацию: один из узлов загружен почти на 100%, в то время как другие на порядок меньше. Алгоритм Round Robin такой ситуации не учитывает в принципе, поэтому перегруженный узел все равно будет получать запросы, потому сфера применения алгоритма Round Robin весьма ограничена.

Weighted Round Robin. Версия алгоритма Round Robin учитывающая весовой коэффициент каждого сервера в соответствии с его производительностью и мощностью, что помогает распределять нагрузку более гибко: серверы с большим весом обрабатывают больше запросов, хотя это всех проблем с отказоустойчивостью не решает.

Кодировка Base64. Зачем нужна?

Base64 — стандарт кодирования двоичных данных при помощи только 64 символов ASCII. Алфавит кодирования содержит латинские символы A-Z, a-z, цифры 0-9 (всего 62 знака) и 2 дополнительных символа, зависящих от системы реализации. Каждые 3 исходных байта кодируются 4-мя символами (увеличение на ⅓).

В формате электронной почты MIME Base64 — это схема, по которой произвольная последовательность байт преобразуется в последовательность печатных ASCII символов.

Стандартные 62 символа дополняют +, / и = — в качестве специального кода суффикса.

Благодаря Base64, в html документы можно включать бинарный контент, создавая единый документ без отдельно расположенных картинок и прочих дополнительных файлов. Таким образом, html документ с включённой в него графикой, аудио, видео, программами, стилями и прочими дополнениями становится прекрасной альтернативой другим форматам сложно оформленных документов типа doc, docx, pdf.

Некоторые приложения кодируют двоичные данные для удобства включения в URL, скрытые поля форм.

АЛГОРИТМ КОДИРОВКИ!!!!!

https://flash2048.com/post/base64