**SPX**

SPX (Sequenced Packet Exchange) і його вдосконалена модифікація SPX II представляють собою транспортні протоколи 7-рівневої моделі OSI. Це протокол гарантує доставку пакета і використовує техніку ковзаючого вікна (віддалений аналог протоколу TCP). У разі втрати або помилки пакет пересилається повторно, число повторень задається програмно. У протоколі SPX не передбачена широкомовна або мультикастинг-адресація. У SPX індукується ситуація, коли партнер несподівано перериває з'єднання, наприклад через обрив зв'язку. Пакети SPX вкладаються в пакети IPX\*. При цьому в полі тип пакету IPX записується код 5. Заголовок пакета SPX завжди містить 42 байти, включаючи 30 байт заголовка IPX-пакета, куди він вкладений.

*\*IPX (англ. Internetwork Packet Exchange) — протокол мережного рівня моделі OSI, призначений для передачі дейтаграм у системах, неорієнтованих на з'єднання (також як й IP або NetBIOS, розроблений IBM й емульований в Novell), він забезпечує зв'язок між NetWare-серверами й кінцевими станціями.*

*Оригінальний транспортний протокол Novell не сприяє успіху цієї мережі. Не встигнувши вчасно переорієнтуватися на транспортні й маршрутні протоколи стека TCP/IP цей украй популярний зовсім недавно вид мереж у цей час має шанси зникнути.*

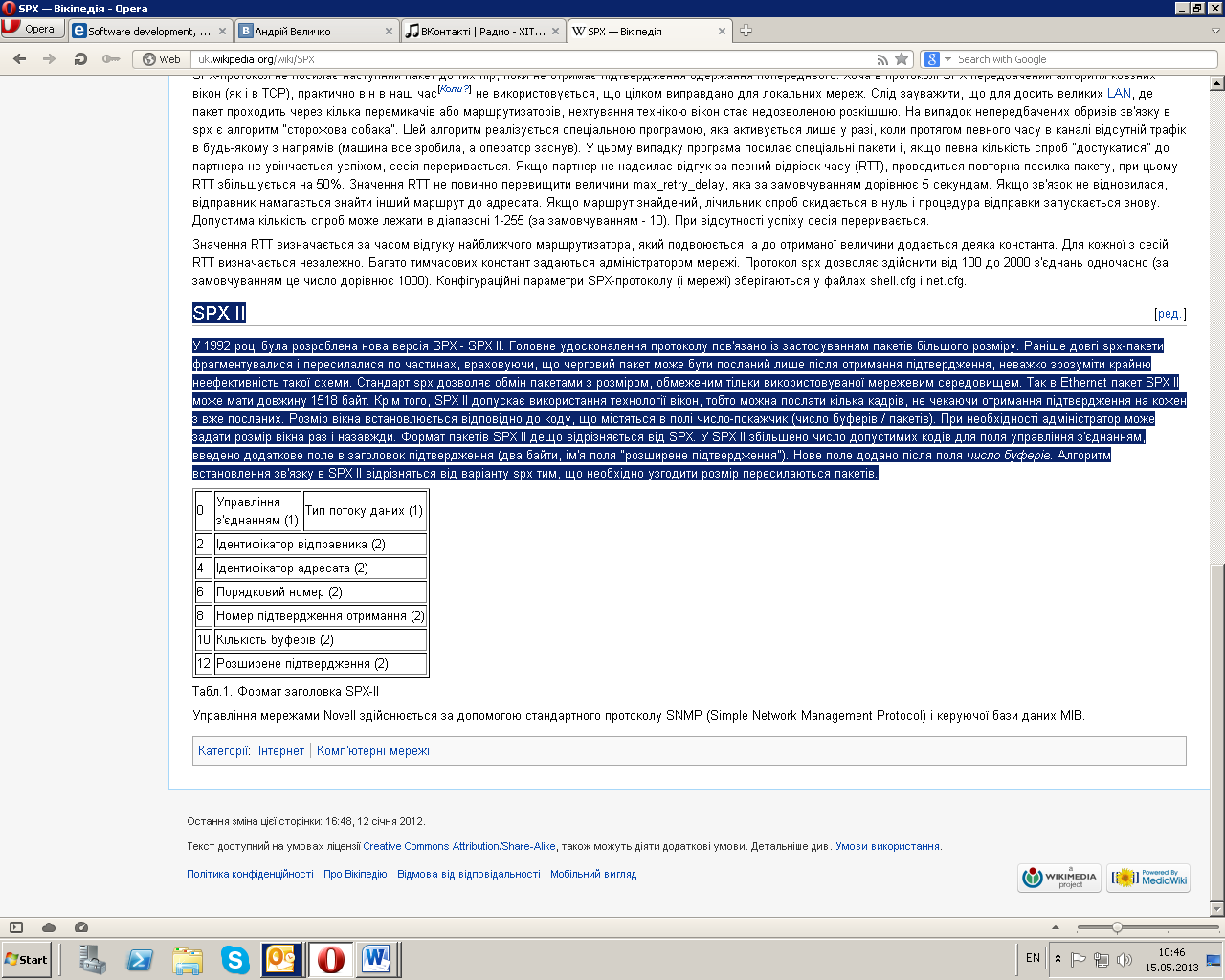
*Алгоритм надсилання пакетів*

SPX-протокол не посилає наступний пакет до тих пір, поки не отримає підтвердження одержання попереднього. Хоча в протоколі SPX передбачений алгоритм ковзних вікон (як і в TCP), практично він в наш час[Коли?] не використовується, що цілком виправдано для локальних мереж. Слід зауважити, що для досить великих LAN, де пакет проходить через кілька перемикачів або маршрутизаторів, нехтування технікою вікон стає недозволеною розкішшю. На випадок непередбачених обривів зв'язку в spx є алгоритм "сторожова собака". Цей алгоритм реалізується спеціальною програмою, яка активується лише у разі, коли протягом певного часу в каналі відсутній трафік в будь-якому з напрямів (машина все зробила, а оператор заснув). У цьому випадку програма посилає спеціальні пакети і, якщо певна кількість спроб "достукатися" до партнера не увінчається успіхом, сесія переривається. Якщо партнер не надсилає відгук за певний відрізок часу (RTT), проводиться повторна посилка пакету, при цьому RTT збільшується на 50%. Значення RTT не повинно перевищити величини max\_retry\_delay, яка за замовчуванням дорівнює 5 секундам. Якщо зв'язок не відновилася, відправник намагається знайти інший маршрут до адресата. Якщо маршрут знайдений, лічильник спроб скидається в нуль і процедура відправки запускається знову. Допустима кількість спроб може лежати в діапазоні 1-255 (за замовчуванням - 10). При відсутності успіху сесія переривається.

Значення RTT визначається за часом відгуку найближчого маршрутизатора, який подвоюється, а до отриманої величини додається деяка константа. Для кожної з сесій RTT визначається незалежно. Багато тимчасових констант задаються адміністратором мережі. Протокол spx дозволяє здійснити від 100 до 2000 з'єднань одночасно (за замовчуванням це число дорівнює 1000). Конфігураційні параметри SPX-протоколу (і мережі) зберігаються у файлах shell.cfg і net.cfg.

*SPX II*

У 1992 році була розроблена нова версія SPX - SPX II. Головне удосконалення протоколу пов'язано із застосуванням пакетів більшого розміру. Раніше довгі spx-пакети фрагментувалися і пересилалися по частинах, враховуючи, що черговий пакет може бути посланий лише після отримання підтвердження, неважко зрозуміти крайню неефективність такої схеми. Стандарт spx дозволяє обмін пакетами з розміром, обмеженим тільки використовуваної мережевим середовищем. Так в Ethernet пакет SPX II може мати довжину 1518 байт. Крім того, SPX II допускає використання технології вікон, тобто можна послати кілька кадрів, не чекаючи отримання підтвердження на кожен з вже посланих. Розмір вікна встановлюється відповідно до коду, що містяться в полі число-покажчик (число буферів / пакетів). При необхідності адміністратор може задати розмір вікна раз і назавжди. Формат пакетів SPX II дещо відрізняється від SPX. У SPX II збільшено число допустимих кодів для поля управління з'єднанням, введено додаткове поле в заголовок підтвердження (два байти, ім'я поля "розширене підтвердження"). Нове поле додано після поля число буферів. Алгоритм встановлення зв'язку в SPX II відрізняться від варіанту spx тим, що необхідно узгодити розмір пересилаються пакетів.



Управління мережами Novell здійснюється за допомогою стандартного протоколу SNMP (Simple Network Management Protocol) і керуючої бази даних MIB.

**FTP**

Протокол передачі файлів (англ. File Transfer Protocol, FTP) — дає можливість абоненту обмінюватися двійковими і текстовими файлами з будь-яким комп'ютером мережі, що підтримує протокол FTP. Установивши зв'язок з віддаленим комп'ютером, користувач може скопіювати файл з віддаленого комп'ютера на свій, або скопіювати файл з свого комп'ютера на віддалений.

При розгляді FTP як сервісу Інтернет мають на увазі не просто протокол, а саме сервіс — доступ до файлів, які знаходяться у файлових архівах.

FTP — стандартна програма, яка працює за протоколом TCP, яка завжди поставляється з операційною системою. Її початкове призначення — передача файлів між різними комп'ютерами, які працюють у мережах TCP/IP: на одному з комп'ютерів працює програма-сервер, на іншому — програма-клієнт, запущена користувачем, яка з'єднується з сервером і передає або отримує файли через FTP-сервіс. Все це розглядається з припущенням, що користувач зареєстрований на сервері та використовує логін та пароль на цьому комп'ютері.

Ця риса послужила причиною того, що програми FTP стали частиною окремого сервісу Інтернету. Справа в тому, що доволі часто сервер FTP налаштовується таким чином, що з'єднатися з ним можна не тільки під своїм ім'ям, але й під умовним іменем anonymous — анонім. У такому випадку для користувача стає доступною не вся файлова система комп'ютера, а лише деякий набір файлів на сервері, які складають вміст серверу anonymous FTP — публічного файлового архіву. Отже, якщо користувач хоче надати у вільне користування файли з інформацією, програмами і т. і., то йому достатньо організувати на власному комп'ютері, включеному в Інтернет, сервер anonymous FTP. Створення такого серверу — процес доволі простий, програми-клієнти FTP вельми розповсюджені, — тому сьогодні публічні файлові архіви організовані в основному як сервери anonymous FTP. Перелік інформації, яка міститься на таких серверах, включає всі аспекти життя: від звичайних текстів до мультимедіа.

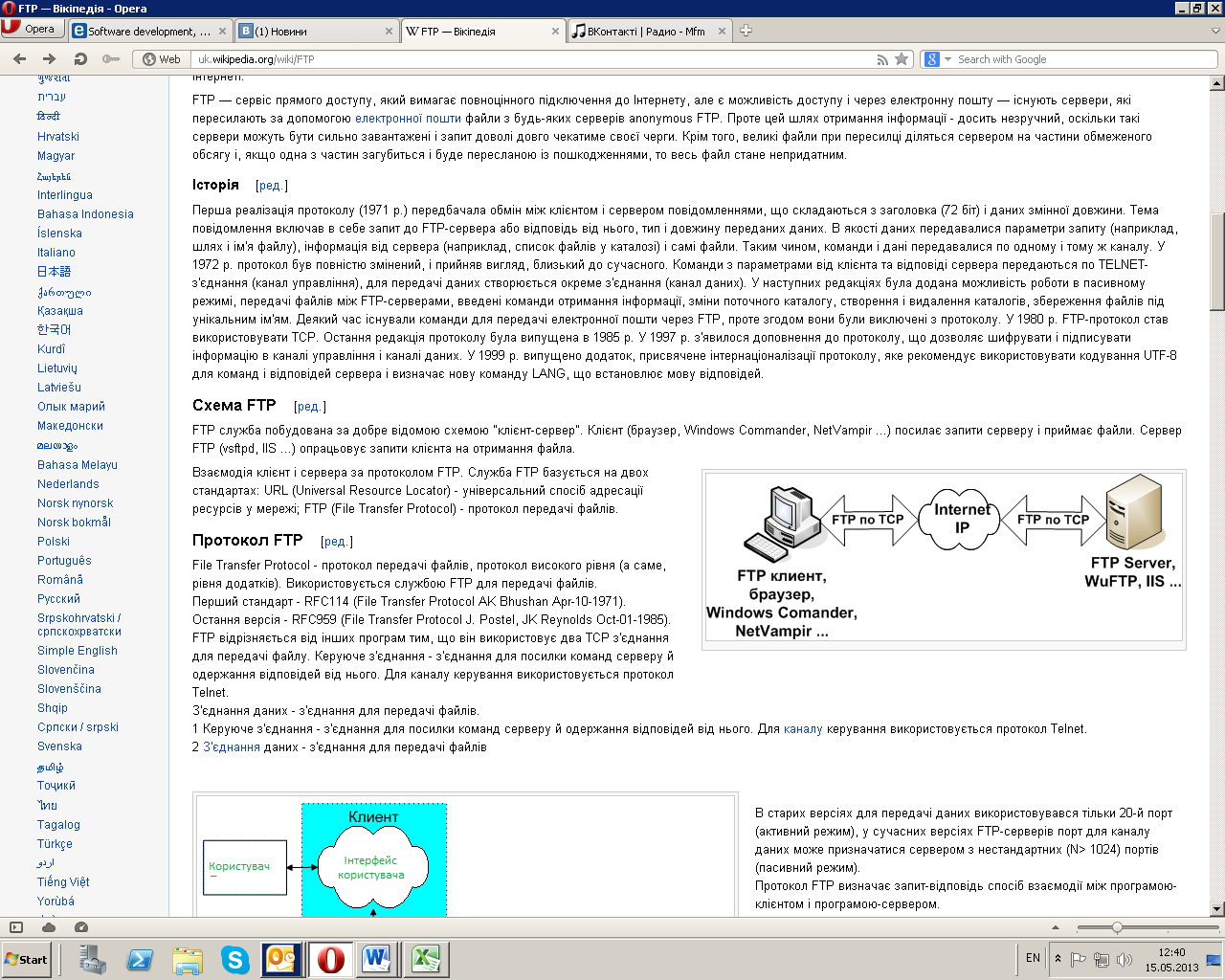
Не зважаючи на розповсюдженість, у FTP є багато недоліків. Програми-клієнти FTP не завжди зручні і прості у користуванні. Користувач не завжди може зрозуміти який файл перед ним, чи той що необхідно, чи ні. Окрім того, не існує простого і універсального засобу для пошуку на серверах anonymous FTP, — хоча для цього і існує спеціальний сервіс archie, але це незалежна програма, вона не універсальна і не завжди її можна ефективно застосовувати. Програми FTP доволі старі і деякі їхні особливості, які були потрібні в часи їхнього створення, не зовсім зрозумілі і потрібні зараз. Наприклад, для передачі файлів існує два режими — двійковий та текстовий, і, якщо користувач неправильно обрав режим передачі, то файл, який необхідно передати, може бути пошкодженим. Опис файлів на сервері видається у форматі операційної системи серверу, а список файлів операційної системи UNIX не завжди з розумінням сприймається користувачами DOS. Сервери FTP нецентралізовані, — звідси випливають ще деякі проблеми. Але незважаючи на все це, сервери anonymous FTP сьогодні — це стандартний шлях організації публічних файлових архівів в Інтернеті.

FTP — сервіс прямого доступу, який вимагає повноцінного підключення до Інтернету, але є можливість доступу і через електронну пошту — існують сервери, які пересилають за допомогою електронної пошти файли з будь-яких серверів anonymous FTP. Проте цей шлях отримання інформації - досить незручний, оскільки такі сервери можуть бути сильно завантажені і запит доволі довго чекатиме своєї черги. Крім того, великі файли при пересилці діляться сервером на частини обмеженого обсягу і, якщо одна з частин загубиться і буде пересланою із пошкодженнями, то весь файл стане непридатним.

*Схема FTP [ред.]*

FTP служба побудована за добре відомою схемою "клієнт-сервер". Клієнт (браузер, Windows Commander, NetVampir ...) посилає запити серверу і приймає файли. Сервер FTP (vsftpd, IIS ...) опрацьовує запити клієнта на отримання файла.

Взаємодія клієнт і сервера за протоколом FTP. Служба FTP базується на двох стандартах: URL (Universal Resource Locator) - універсальний спосіб адресації ресурсів у мережі; FTP (File Transfer Protocol) - протокол передачі файлів.



*Протокол FTP [ред.]*

File Transfer Protocol - протокол передачі файлів, протокол високого рівня (а саме, рівня додатків). Використовується службою FTP для передачі файлів.

Перший стандарт - RFC114 (File Transfer Protocol AK Bhushan Apr-10-1971).

Остання версія - RFC959 (File Transfer Protocol J. Postel, JK Reynolds Oct-01-1985).

FTP відрізняється від інших програм тим, що він використовує два TCP з'єднання для передачі файлу. Керуюче з'єднання - з'єднання для посилки команд серверу й одержання відповідей від нього. ***Для каналу керування використовується протокол Telnet.***

З'єднання даних - з'єднання для передачі файлів.

1 Керуюче з'єднання - з'єднання для посилки команд серверу й одержання відповідей від нього. Для каналу керування використовується протокол Telnet.

2 З'єднання даних - з'єднання для передачі файлів

В старих версіях для передачі даних використовувався тільки 20-й порт (активний режим), у сучасних версіях FTP-серверів порт для каналу даних може призначатися сервером з нестандартних (N> 1024) портів (пасивний режим).

Протокол FTP визначає запит-відповідь спосіб взаємодії між програмою-клієнтом і програмою-сервером.

Робота FTP на рівні користувача містить кілька етапів:

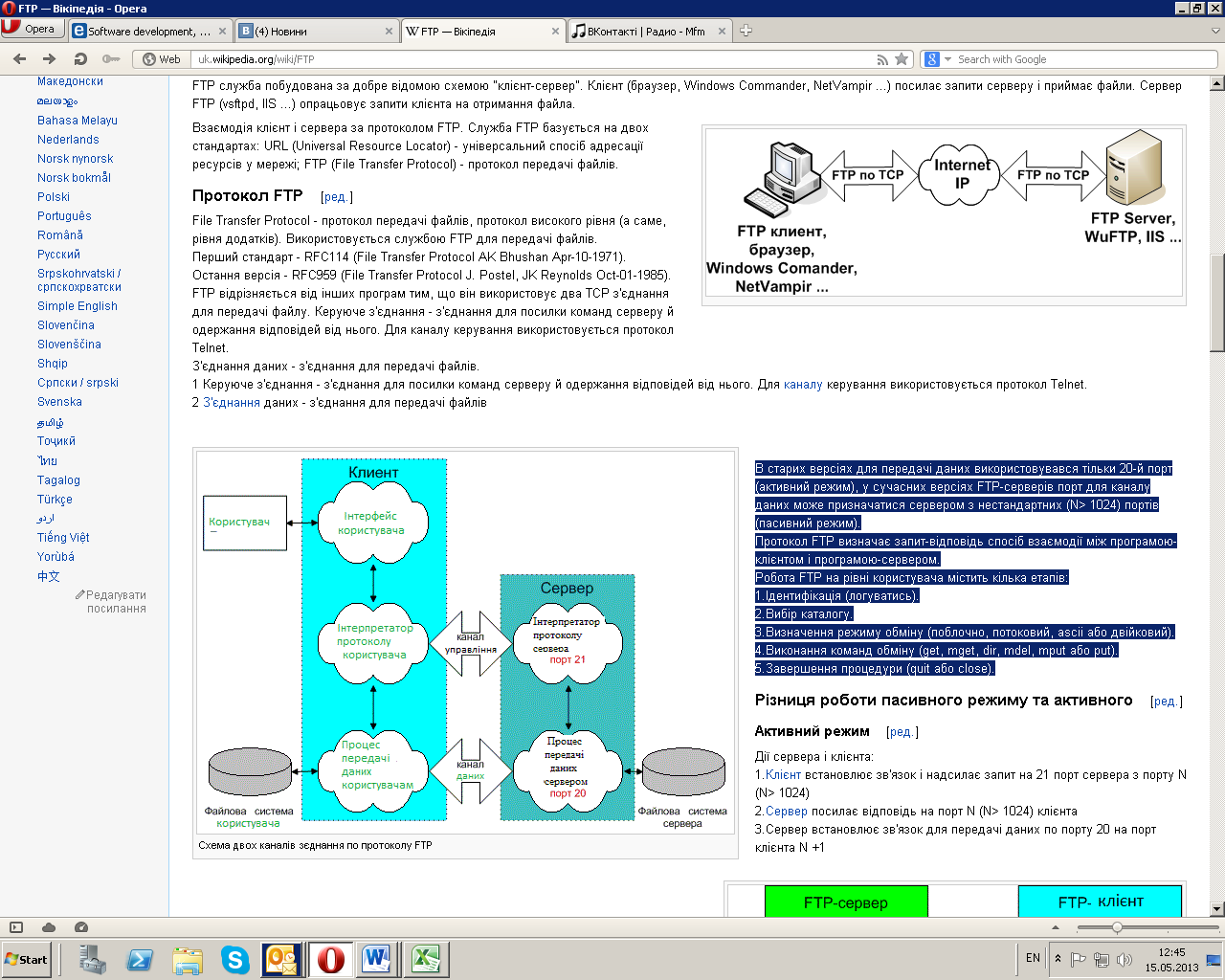
1.Ідентифікація (логуватись).

2.Вибір каталогу.

3.Визначення режиму обміну (поблочно, потоковий, ascii або двійковий).

4.Виконання команд обміну (get, mget, dir, mdel, mput або put).

5.Завершення процедури (quit або close).



*Різниця роботи пасивного режиму та активного [ред.]*

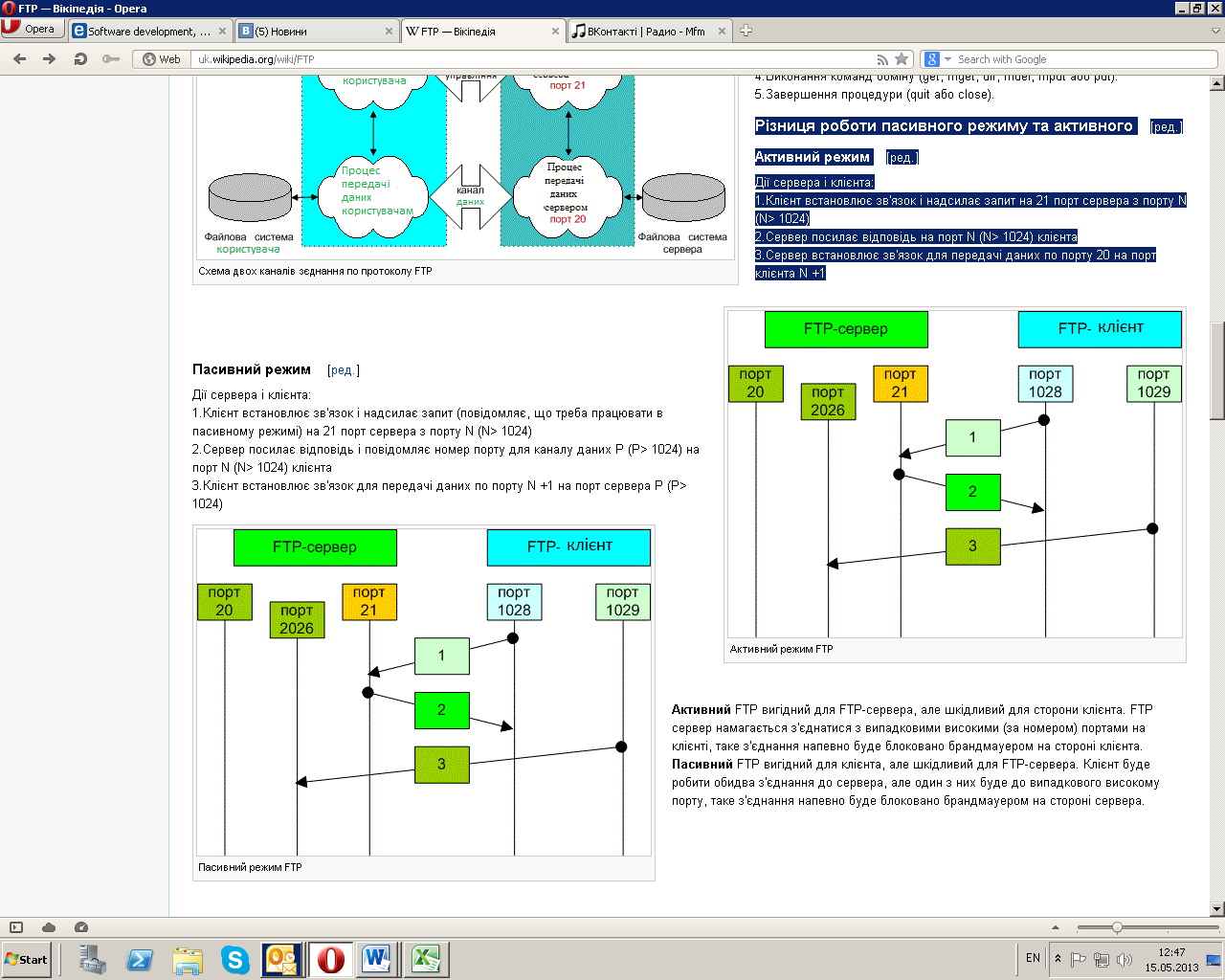
*Активний режим [ред.]*

Дії сервера і клієнта:

1.Клієнт встановлює зв'язок і надсилає запит на 21 порт сервера з порту N (N> 1024)

2.Сервер посилає відповідь на порт N (N> 1024) клієнта

3.Сервер встановлює зв'язок для передачі даних по порту 20 на порт клієнта N +1



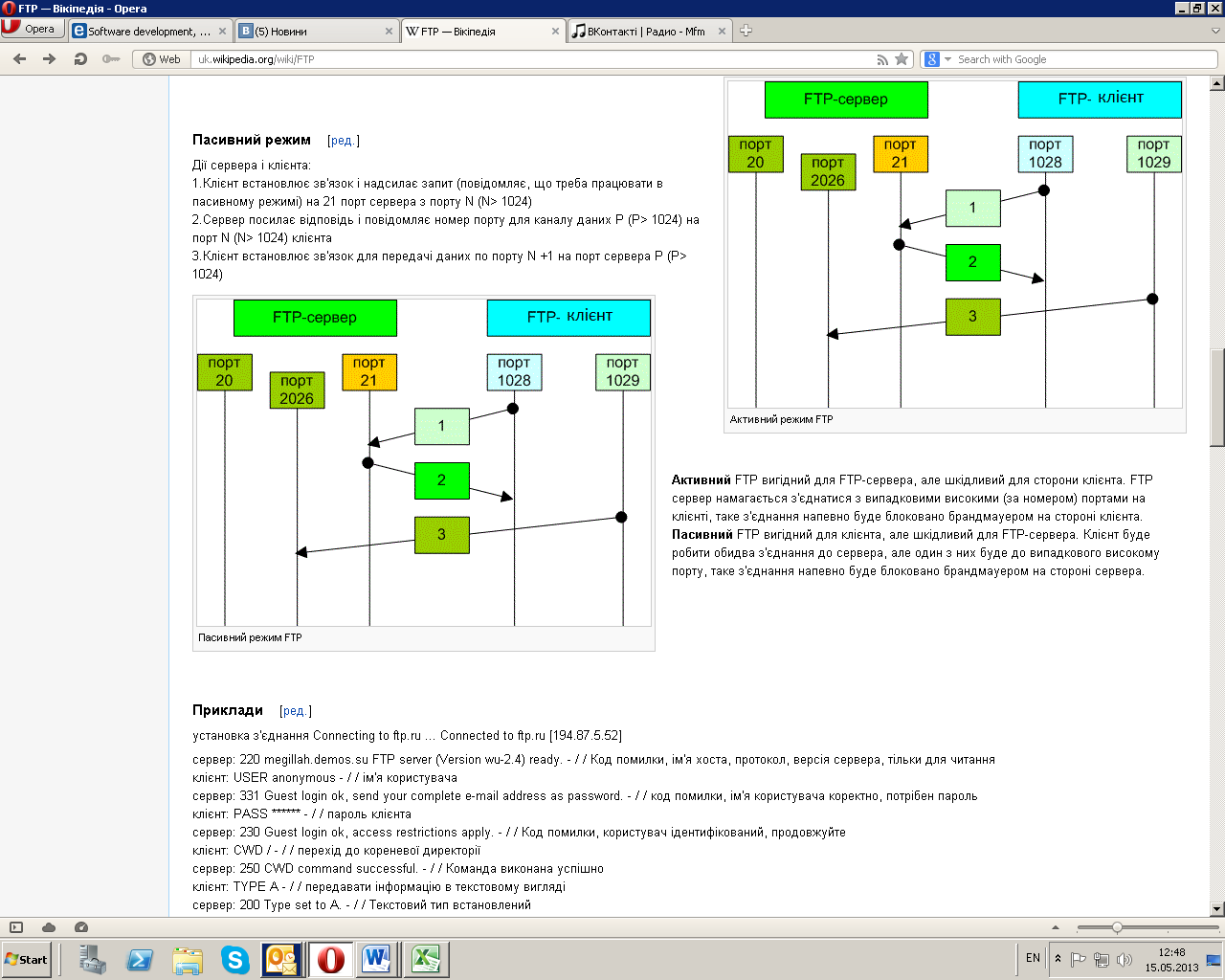
*Пасивний режим [ред.]*

Дії сервера і клієнта:

1.Клієнт встановлює зв'язок і надсилає запит (повідомляє, що треба працювати в пасивному режимі) на 21 порт сервера з порту N (N> 1024)

2.Сервер посилає відповідь і повідомляє номер порту для каналу даних P (P> 1024) на порт N (N> 1024) клієнта

3.Клієнт встановлює зв'язок для передачі даних по порту N +1 на порт сервера P (P> 1024)



Активний FTP вигідний для FTP-сервера, але шкідливий для сторони клієнта. FTP сервер намагається з'єднатися з випадковими високими (за номером) портами на клієнті, таке з'єднання напевно буде блоковано брандмауером на стороні клієнта.

Пасивний FTP вигідний для клієнта, але шкідливий для FTP-сервера. Клієнт буде робити обидва з'єднання до сервера, але один з них буде до випадкового високому порту, таке з'єднання напевно буде блоковано брандмауером на стороні сервера.

**NTP**

Мережевий протокол часу (англ. Network Time Protocol) — мережевий протокол синхронізації внутрішнього годинника комп'ютера з використанням мереж зі змінною латентністю, заснований на комутації пакетів.

Хоча традиційно NTP використає для своєї роботи протокол UDP, він також здатний працювати й поверх TCP. Система NTP надзвичайно стійка до змін латентності середовища передачі.

NTP використовує алгоритм Марзулло (запропонований Кейтом Марзулло (Keith Marzullo) з Університету Каліфорнії, Сан-Дієго), включаючи таку особливість, як облік часу передачі. У версії 4 здатний досягати точності 10 мс (1/100 с) при роботі через Інтернет, і до 200 мікросекунд (1/5000 с) і краще усередині локальних мереж.

NTP — один з найстарших використовуваних протоколів. NTP розроблений Девідом Л. Міллсом (David L. Mills) з університету Делавера й у цей час продовжує вдосконалювання. Поточна версія — NTP 4.

Демон NTP — програма рівня користувача, працює постійно.

NTP використає ієрархічну систему «годинникових рівнів», рівень 1 синхронізований з високоточними годинниками, наприклад, система GPS або еталони часу. NTP рівень 2 синхронізується з однієї з машин рівня 1, і так далі.

Час, представляється в системі NTP 64-бітним числом, що складається з 32-бітного лічильника секунд й 32-бітного лічильника часток секунди, дозволяючи передавати час у діапазоні 232 секунд, з теоретичною точністю 2−32 секунди. Оскільки шкала часу в NTP повторюється кожні 232 секунди (136 років), одержувач повинен хоча б приблизно знати поточний час (з точністю 50 років).

Простіша реалізація цього алгоритму відома як SNTP — простий синхронізуючий мережний протокол. Використовується у вбудованих системах, і пристроях, що не вимагають високої точності.

*NTP использует иерархическую, многоуровневую систему источников времени.* Каждый уровень этой иерархии называется слоем, каждому слою присваивается номер, начиная с 0 (ноль) в верхней части. Уровень слоя определяет расстояние от эталонных часов и существует, чтобы предотвратить циклические зависимости в иерархии. Важно отметить, что слой не является показателем качества и надежности, это значит, что источник слоя 3 может дать сигнал более высокого качества, чем некоторые источники слоя 2. В основном, слои служат для распределения нагрузки и обеспечения большей площади покрытия. Это определение слоя также отличается от понятия часовых слоёв, используемых в телекоммуникационных системах.

Слой 0

Слой 0 - это высокоточные приборы служащие эталоном времени, такие как атомные (молекулярные, квантовые) часы, радиочасы или их аналоги. Обычно эти устройства не подключены к сети; вместо этого они подключены к локальному компьютеру (например, через интерфейс RS-232) и передают сигналы PPS для синхронизации.

Слой 1

Это компьютер, к которому напрямую подключены эталонные часы. Он выступает в качестве сетевого сервера времени и отвечает на NTP-запросы посылаемые компьютерами слоя 2.

Слой 2

Это компьютеры, которые получают время от серверов первого слоя, используя для этого протокол NTP. Обычно, компьютеры второго слоя обращаются к нескольким серверам первого слоя, и используя NTP-алгоритм, получают наилучший образец данных, отсеивая сервера с очевидно неверным временем. Компьютеры могут сравнивать свои данные с другими компьютерами своего слоя для получения стабильных и непротиворечивых данных на всех компьютерах слоя. Компьютеры второго слоя в свою очередь выступают в качестве серверов для компьютеров третьего слоя и отвечают на NTP-запросы.

Слой 3

Компьютеры третьего слоя работают точно так же как и компьютеры второго слоя, с той лишь разницей, что серверами для них являются компьютеры вышележащего второго слоя. Они так же могут выступать в качестве серверов для нижележащего слоя. NTP (в зависимости от версии) поддерживает до 256 слоев.

**TELNET**

TELNET (англ. TErminaL NETwork) — сетевой протокол для реализации текстового интерфейса по сети (в современной форме — при помощи транспорта TCP). Название «telnet» имеют также некоторые утилиты, реализующие клиентскую часть протокола. Современный стандарт протокола описан в RFC 854.

Выполняет функции протокола прикладного уровня модели OSI.

*Применения[править]*

Исторически Telnet служил для удалённого доступа к интерфейсу командной строки операционных систем. Впоследствии его стали использовать для прочих текстовых интерфейсов, вплоть до игр MUD и анимированного ASCII-art. Теоретически, даже обе стороны протокола могут являться не только людьми, но и программами.

Иногда клиенты telnet используются для доступа к другим протоколам на основе транспорта TCP, см. Telnet и другие протоколы.

Протокол telnet используется в управляющем соединении FTP, то есть заходить на сервер командой telnet ftp.example.net ftp для выполнения отладки и экспериментов не только возможно, но и правильно (в отличие от применения клиентов telnet для доступа к HTTP, IRC и большинству других протоколов).

*Безопасность[править]*

В протоколе не предусмотрено использование ни шифрования, ни проверки подлинности данных. Поэтому он уязвим для любого вида атак, к которым уязвим его транспорт, то есть протокол TCP. Для функциональности удалённого доступа к системе в настоящее время применяется сетевой протокол SSH (особенно его версия 2), при создании которого упор делался именно на вопросы безопасности. Так что следует иметь в виду, что сессия Telnet весьма беззащитна, если только не осуществляется в полностью контролируемой сети или с применением защиты на сетевом уровне (различные реализации виртуальных частных сетей). По причине ненадёжности от Telnet как средства управления операционными системами давно отказались.

**SSH**

SSH (англ. Secure SHell — «безопасная оболочка»[1]) — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов). Схож по функциональности с протоколами Telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли. SSH допускает выбор различных алгоритмов шифрования. SSH-клиенты и SSH-серверы доступны для большинства сетевых операционных систем.

SSH позволяет безопасно передавать в незащищённой среде практически любой другой сетевой протокол. Таким образом, можно не только удалённо работать на компьютере через командную оболочку, но и передавать по шифрованному каналу звуковой поток или видео (например, с веб-камеры)[2]. Также SSH может использовать сжатие передаваемых данных для последующего их шифрования, что удобно, например, для удалённого запуска клиентов X Window System.

Большинство хостинг-провайдеров за определённую плату предоставляют клиентам доступ к их домашнему каталогу по SSH. Это может быть удобно как для работы в командной строке, так и для удалённого запуска программ (в том числе графических приложений).

**IMAP**

IMAP (англ. Internet Message Access Protocol) — протокол прикладного уровня для доступа к электронной почте.

Базируется на транспортном протоколе TCP и использует порт 143.

IMAP предоставляет пользователю обширные возможности для работы с почтовыми ящиками, находящимися на центральном сервере. Почтовая программа, использующая этот протокол, получает доступ к хранилищу корреспонденции на сервере так, как будто эта корреспонденция расположена на компьютере получателя. Электронными письмами можно манипулировать с компьютера пользователя (клиента) без постоянной пересылки с сервера и обратно файлов с полным содержанием писем.

Для отправки писем используется протокол SMTP.

*Цель разработки протокола IMAP[править]*

Протокол IMAP представляет собой альтернативу POP3.

POP3 имеет ряд недостатков, и наиболее серьёзный из них — отсутствие возможностей по управлению перемещением и хранением сообщений на сервере. Сообщения, как правило, загружаются с почтового сервера все сразу, после чего они с сервера удаляются, то есть отсутствует возможность выбирать сообщения для получения.

Для решения проблем, связанных с этой особенностью POP3, в Вашингтонском университете был разработан новый протокол, предполагающий возможность получения пользователями электронной почты из одного почтового ящика из различных мест, при этом сообщения не распределяются между точками получения. Пользователю предоставляется возможность управлять сообщениями в его почтовом ящике и дополнительными функциями по обслуживанию почтовых ящиков на сервере.

*Преимущества по сравнению с POP3[править]*

При использовании POP3 клиент подключается к серверу только на промежуток времени, необходимый для загрузки новых сообщений. При использовании IMAP соединение не разрывается, пока пользовательский интерфейс активен, а сообщения загружаются только по требованию клиента. Это позволяет уменьшить время отклика для пользователей, в чьих ящиках имеется много сообщений большого объёма.

Протокол POP требует, чтоб текущий клиент был единственным подключенным к ящику. IMAP позволяет одновременный доступ нескольких клиентов к ящику и предоставляет клиенту возможность отслеживать изменения, вносимые другими клиентами, подключенными одновременно с ним.

Благодаря системе флагов, определенной в IMAP4, клиент может отслеживать состояние сообщения (прочитано, отправлен ответ, удалено и т. д.); данные о флагах хранятся на сервере.

Клиенты IMAP4 могут создавать, переименовывать и удалять ящики и перемещать сообщения между ящиками. Кроме того, можно использовать расширение IMAP4 Access Control List (ACL) Extension (RFC 4314) для управления правами доступа к ящикам.

Поиск сообщений происходит на стороне сервера.

IMAP4 имеет явный механизм расширения.

**POP3**

POP3 (англ. Post Office Protocol Version 3 — протокол почтового отделения, версия 3) — стандартный Интернет-протокол прикладного уровня, используемый клиентами электронной почты для извлечения электронного сообщения с удаленного сервера по TCP/IP-соединению.

POP и IMAP (Internet Message Access Protocol) — наиболее распространенные Интернет-протоколы для извлечения почты. Практически все современные клиенты и серверы электронной почты поддерживают оба стандарта. Протокол POP был разработан в нескольких версиях, нынешним стандартом является третья версия (POP3). Большинство поставщиков услуг электронной почты (такие как Hotmail, Gmail и Yahoo! Mail) также поддерживают IMAP и POP3. Предыдущие версии протокола (POP, POP2) устарели.

Альтернативным протоколом для сбора сообщений с почтового сервера является IMAP.

*Общие сведения[править]*

POP поддерживает простые требования «загрузи-и-удали» для доступа к удаленным почтовым ящикам. Хотя большая часть POP-клиентов предоставляют возможность оставить почту на сервере после загрузки, использующие POP клиенты обычно соединяются, извлекают все письма, сохраняют их на пользовательском компьютере как новые сообщения, удаляют их с сервера, после чего разъединяются.

Другие протоколы, в частности IMAP, предоставляют более полный и комплексный удаленный доступ к типичным операциям с почтовым ящиком. Многие клиенты электронной почты поддерживают как POP, так и IMAP; однако, гораздо меньше интернет-провайдеров поддерживают IMAP.

POP3-сервер прослушивает общеизвестный порт 110. Шифрование связи для POP3 запрашивается после запуска протокола, с помощью либо команды STLS (если она поддерживается), либо POP3S, которая соединяется с сервером используя TLS или SSL по TCP-порту 995.

Доступные сообщения клиента фиксируются при открытии почтового ящика POP-сессией и определяются количеством сообщений для сессии, или, по желанию, с помощью уникального идентификатора, присваиваемого сообщению POP-сервером. Этот уникальный идентификатор является постоянным и уникальным для почтового ящика и позволяет клиенту получить доступ к одному и тому же сообщению в разных POP-сессиях. Почта извлекается и помечается для удаления с помощью номера сообщения. При выходе клиента из сессии помеченные сообщения удаляются из почтового ящика.

*Расширения[править]*

Механизм расширения был предложен в RFC 2449 для размещения новых расширений, а также организованного объявления о поддержке опциональных команд, таких как TOP и UIDL. RFC не намеревались поощрять расширения и подтвердили, что роль POP3 заключается в предоставлении простой поддержки в основном для требования «загрузи-и-удали».

Расширения выводятся списком командой CAPA. За исключением APOP, все опциональные команды были включены в изначальный набор возможностей. Вслед за ESMTP (RFC 5321), возможности, начинающиеся с "X.

*Сравнение с IMAP[править]*

Клиенты, которые оставляют почту на серверах, обыкновенно используют команду UIDL для получения текущего соответствия между количеством сообщений и сообщением, определяемым его уникальным идентификатором. Идентификатор произволен и может повторяться, если на ящике есть идентичные сообщения. Напротив, IMAP использует 32-битный уникальный идентификатор (UID), присваиваемый сообщениям по возрастанию (но не обязательно подряд) по мере их получения. При извлечении новых сообщений IMAP-клиенты запрашивают UID больший, чем наивысшее значение UID среди всех ранее извлеченных сообщений, в то время как POP-клиент должен выбирать из всей карты UIDL. Для больших почтовых ящиков это может потребовать значительной обработки.

MIME служит в качестве стандарта для вложений и не-ASCII текста в электронных сообщениях. Хотя ни POP3, ни SMTP не требуют MIME-отформатированного сообщения, по существу, все не-ASCII сообщения идут в формате MIME, поэтому POP-клиенты должны также «понимать» и использовать MIME. IMAP, по определению, принимает MIME-форматированные сообщения.

**SMTP**

SMTP (англ. Simple Mail Transfer Protocol — простой протокол передачи почты) — это широко используемый сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP.

SMTP впервые был описан в RFC 821 (1982 год); последнее обновление в RFC 5321 (2008) включает масштабируемое расширение — ESMTP (англ. Extended SMTP). В настоящее время под «протоколом SMTP», как правило, подразумевают и его расширения. Протокол SMTP предназначен для передачи исходящей почты с использованием порта TCP 25.

В то время, как электронные почтовые серверы и другие агенты пересылки сообщений используют SMTP для отправки и получения почтовых сообщений, работающие на пользовательском уровне клиентские почтовые приложения обычно используют SMTP только для отправки сообщений на почтовый сервер для ретрансляции. Для получения сообщений клиентские приложения обычно используют либо POP (англ. Post Office Protocol — протокол почтового отделения), либо IMAP (англ. Internet Message Access Protocol), либо патентованные системы (такие как Microsoft Exchange и Lotus Notes/Domino) для доступа к учетной записи своего почтового ящика на сервере.

*История[править]*

Стандарт SMTP был разработан примерно в то же время, что и Usenet, сеть передачи данных, имеющая некоторые сходства с SMTP. SMTP стал широко использоваться в ранние 1980-е. В то время он был дополнением для работающей под Unix почтовой программы Unix Copy Program (UUCP), которая больше подходила для обработки передачи электронных сообщений между периодически связанными устройствами. С другой стороны, SMTP прекрасно работает, когда как отправляющее, так и принимающее устройства связаны в сети постоянно. Оба устройства используют механизм хранения и пересылки и являются примером push-технологии (технологии «проталкивания»). Хотя новостные группы Usenet все еще распространяются между серверами с помощью UUCP, почта UUCP фактически исчезла вместе с маршрутом «bang path» (последовательность хост-машин в сети, по которой сообщение должно дойти до адресата), которые использовались как заголовки маршрутизации. В статье о перезаписи отправителя содержится техническая справочная информация о истории раннего SMTP и маршрутизации от источника до RFC 1123.

Sendmail был одним из первых (если не первым) агентом пересылки сообщений, в котором был реализован SMTP. В число других популярных серверных программ, поддерживающих SMTP, входят Postfix, qmail, Novell GroupWise, Exim, Novell NetMail, Microsoft Exchange Server, Sun Java System Messaging Server.

Предоставление сообщений (RFC 2476) и SMTP-AUTH (RFC 2554) были введены в 1998 и 1999 г.г. и описывали новые тенденции в передаче электронных сообщений. Изначально, SMTP-сервера были обычно внутренними для организации, получая сообщения от организаций извне и ретранслируя сообщения организации во внешнюю среду. Но с течением времени, SMTP-сервера (агенты пересылки сообщений), на деле, расширяли свои функции и в конце концов стали агентами предоставления сообщений для пользовательских почтовых приложений, некоторые из которых теперь ретранслировали почту извне организации (например, руководитель компании, будучи в поездке, хочет отправить электронное сообщение с помощью корпоративного SMTP-сервера).

Данный вопрос, являясь следствием быстрого развития и популярности Всемирной паутины, означает, что SMTP должен был включать в себя особые правила и методы для ретрансляции сообщений и авторизации пользователей для предотвращения таких злоупотреблений, как ретрансляция нежелательной почты (спам).

Поскольку этот протокол сначала был с текстовым (ASCII) интерфейсом, то он плохо работал с бинарными файлами и символами многих неанглийских языков. Такие стандарты, как Multipurpose Internet Mail Extensions (MIME), были разработаны для кодирования двоичных файлов для передачи через SMTP. Разработанные после Sendmail агенты пересылки, как правило, также осуществляли опцию чистых 8 бит, так что альтернативная стратегия «просто посылай восемь» может быть использована для передачи произвольных текстовых данных (в любой восьмибитной ASCII-подобной кодировке символов) через SMTP. Однако все еще оставалась проблема кракозябр, вызванная разным отображением наборов символов у производителей, хотя сами почтовые адреса все еще позволяли использовать исключительно ASCII. Сегодня агенты пересылки, работающие с чистыми 8 битами, как правило, поддерживают расширение 8BITMIME, позволяющее передавать бинарные файлы почти так же легко, как обычный текст. Недавно было создано расширение SMTPUTF8 для поддержки текста в кодировке UTF-8, благодаря чему стало возможным включать международное содержимое и адреса с использованием таких алфавитов, как кириллица или китайский.

*Модель обработки почты*

Синие стрелки могут быть реализованы с использованием различных версий SMTP.

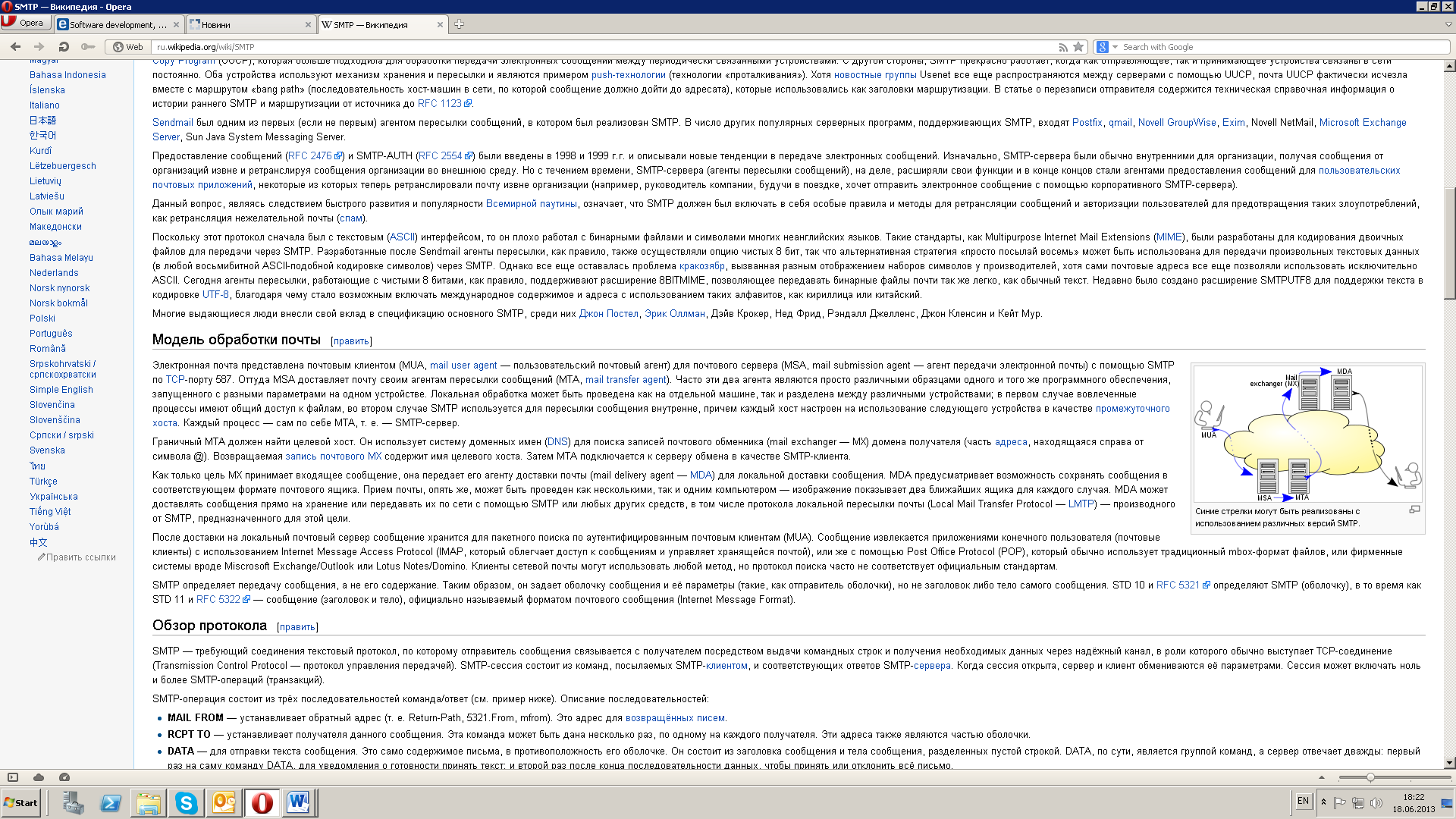
Электронная почта представлена почтовым клиентом (MUA, mail user agent — пользовательский почтовый агент) для почтового сервера (MSA, mail submission agent — агент передачи электронной почты) с помощью SMTP по TCP-порту 587. Оттуда MSA доставляет почту своим агентам пересылки сообщений (MTA, mail transfer agent). Часто эти два агента являются просто различными образцами одного и того же программного обеспечения, запущенного с разными параметрами на одном устройстве. Локальная обработка может быть проведена как на отдельной машине, так и разделена между различными устройствами; в первом случае вовлеченные процессы имеют общий доступ к файлам, во втором случае SMTP используется для пересылки сообщения внутренне, причем каждый хост настроен на использование следующего устройства в качестве промежуточного хоста. Каждый процесс — сам по себе MTA, т. е. — SMTP-сервер.

Граничный MTA должен найти целевой хост. Он использует систему доменных имен (DNS) для поиска записей почтового обменника (mail exchanger — MX) домена получателя (часть адреса, находящаяся справа от символа @). Возвращаемая запись почтового MX содержит имя целевого хоста. Затем MTA подключается к серверу обмена в качестве SMTP-клиента.

Как только цель MX принимает входящее сообщение, она передает его агенту доставки почты (mail delivery agent — MDA) для локальной доставки сообщения. MDA предусматривает возможность сохранять сообщения в соответствующем формате почтового ящика. Прием почты, опять же, может быть проведен как несколькими, так и одним компьютером — изображение показывает два ближайших ящика для каждого случая. MDA может доставлять сообщения прямо на хранение или передавать их по сети с помощью SMTP или любых других средств, в том числе протокола локальной пересылки почты (Local Mail Transfer Protocol — LMTP) — производного от SMTP, предназначенного для этой цели.

После доставки на локальный почтовый сервер сообщение хранится для пакетного поиска по аутентифицированным почтовым клиентам (MUA). Сообщение извлекается приложениями конечного пользователя (почтовые клиенты) с использованием Internet Message Access Protocol (IMAP, который облегчает доступ к сообщениям и управляет хранящейся почтой), или же с помощью Post Office Protocol (POP), который обычно использует традиционный mbox-формат файлов, или фирменные системы вроде Miscrosoft Exchange/Outlook или Lotus Notes/Domino. Клиенты сетевой почты могут использовать любой метод, но протокол поиска часто не соответствует официальным стандартам.

SMTP определяет передачу сообщения, а не его содержание. Таким образом, он задает оболочку сообщения и её параметры (такие, как отправитель оболочки), но не заголовок либо тело самого сообщения. STD 10 и RFC 5321 определяют SMTP (оболочку), в то время как STD 11 и RFC 5322 — сообщение (заголовок и тело), официально называемый форматом почтового сообщения (Internet Message Format).



Msa - агент передачи электронной почты

MTA - агенти пересылки сообщений

(mail exchanger — MX) домена получателя

MDA - агенту доставки почты

*Обзор протокола[править]*

SMTP — требующий соединения текстовый протокол, по которому отправитель сообщения связывается с получателем посредством выдачи командных строк и получения необходимых данных через надёжный канал, в роли которого обычно выступает TCP-соединение (Transmission Control Protocol — протокол управления передачей). SMTP-сессия состоит из команд, посылаемых SMTP-клиентом, и соответствующих ответов SMTP-сервера. Когда сессия открыта, сервер и клиент обмениваются её параметрами. Сессия может включать ноль и более SMTP-операций (транзакций).

SMTP-операция состоит из трёх последовательностей команда/ответ (см. пример ниже). Описание последовательностей:

MAIL FROM — устанавливает обратный адрес (т. е. Return-Path, 5321.From, mfrom). Это адрес для возвращённых писем.

RCPT TO — устанавливает получателя данного сообщения. Эта команда может быть дана несколько раз, по одному на каждого получателя. Эти адреса также являются частью оболочки.

DATA — для отправки текста сообщения. Это само содержимое письма, в противоположность его оболочке. Он состоит из заголовка сообщения и тела сообщения, разделенных пустой строкой. DATA, по сути, является группой команд, а сервер отвечает дважды: первый раз на саму команду DATA, для уведомления о готовности принять текст; и второй раз после конца последовательности данных, чтобы принять или отклонить всё письмо.

Помимо промежуточных ответов для DATA-команды, каждый ответ сервера может быть положительным (код ответа 2хх) или отрицательным. Последний, в свою очередь, может быть постоянным (код 5хх) либо временным (код 4хх). Отказ SMTP-сервера в передаче сообщения — постоянная ошибка; в этом случае клиент должен отправить возвращённое письмо. После сброса — положительного ответа, сообщение скорее всего будет отвержено. Также сервер может сообщить о том, что ожидаются дополнительные данные от клиента (код 3xx).

Изначальным хостом (SMTP-клиентом) может быть как почтовый клиент конечного пользователя (функционально определяемый как почтовый агент — MUA), так и агент пересылки сообщений (MTA) на сервере, т.е. сервер действует как клиент в соответствующей сессии для ретрансляции сообщения. Полностью функциональные сервера поддерживают очереди сообщений для повторной передачи сообщения в случае ошибок.

MUA знает SMTP-сервер для исходящей почты из своих настроек. SMTP-сервер, действующий как клиент, т. е. пересылающий сообщения, определяет, к какому серверу подключиться, просмотром ресурса записей MX (Mail eXchange) DNS для домена каждого получателя. В случае, если запись MX не найдена, совместимые MTA (не все) возвращаются к простой А-записи. Пересылающие сервера также могут быть настроены на использование Smart host.

SMTP-сервер, действующий как клиент, устанавливает TCP-соединение с сервером по разработанному для SMTP порту 25. MUA должен использовать порт 587 для подключения к агенту предоставления сообщений (MSA). Основное различие между MTA и MSA заключается в том, что SMTP-аутентификация обязательно только для последнего.

**MIME**

MIME (произн. «майм», англ. Multipurpose Internet Mail Extensions — многоцелевые расширения интернет-почты) — стандарт, описывающий передачу различных типов данных по электронной почте, а также, шире, спецификация для кодирования информации и форматирования сообщений таким образом, чтобы их можно было пересылать по Интернету.Содержание [убрать]

*Введение[править]*

MIME определяет механизмы для передачи разного рода информации внутри текстовых данных (в частности, с помощью электронной почты), а именно: текст на языках, для которых используются кодировки, отличные от ASCII, и нетекстовый контент, такой как картинки, музыка, фильмы и программы. MIME является также фундаментальным компонентом коммуникационных протоколов, таких как HTTP, которым нужно, чтобы данные передавались в контексте сообщений подобных e-mail, даже если данные реально не являются e-mail.

Основной формат электронных сообщений определен в RFC 5322, который является обновленной версией RFC 2822 (который, в свою очередь, является обновленной версией RFC 822). Эти стандарты определяют похожие форматы для текстовых e-mail-заголовков и содержимого и правил, относящихся к общеиспользуемым полям, таким как To:, Subject:, From: и Date:. MIME определяет набор e-mail-заголовков для определения дополнительных атрибутов сообщения, включая тип контента, и определяет множество кодировок, которые могут быть использованы для представления 8-битных бинарных данных с помощью символов из 7-битного ASCII. MIME также определяет правила для кодирования символов из Extended ASCII (с кодами 128—255) в заголовках e-mail-сообщения, таких как Subject:.

MIME расширяем для новых типов — его определение включает метод для регистрации новых типов контента и других атрибутов.