WIKIPEDIA

BitTorrent

BitTorrent (abbreviated to **BT**) is a communication protocol for peer-to-peer file sharing (P2P) which is used to distribute data and electronic files over the Internet.

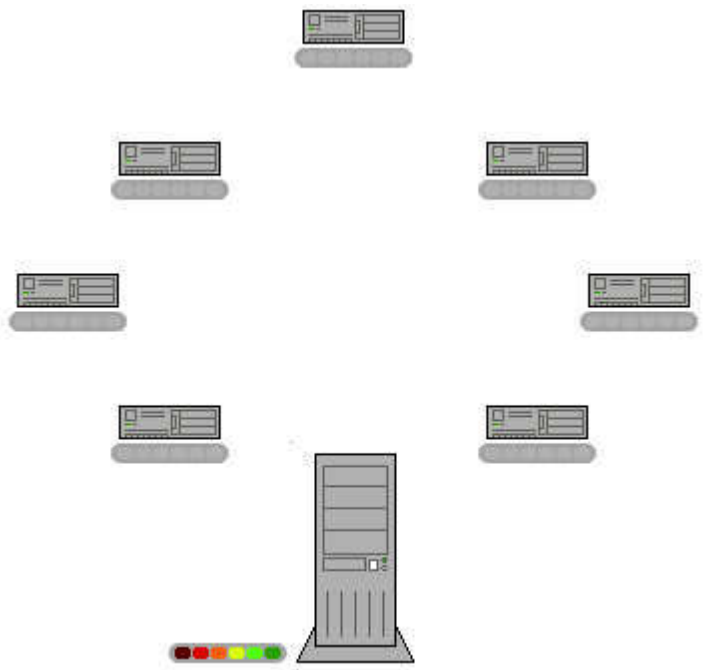
BitTorrent is one of the most common protocols for transferring large files, such as digital video files containing TV shows or video clips or digital audio files containing songs. Peer-to-peer networks have been estimated to collectively account for approximately 43% to 70% of all Internet traffic (depending on location) as of February 2009.[1] In February 2013, BitTorrent was responsible for 3.35% of all worldwide bandwidth, more than half of the 6% of total bandwidth dedicated to file sharing.[2]

To send or receive files, a person uses a BitTorrent client on their Internet-connected computer. A BitTorrent client is a computer program that implements the BitTorrent protocol. Popular clients include µTorrent, Xunlei,[3] Transmission, qBittorrent, Vuze, Deluge, BitComet and Tixati. BitTorrent trackers provide a list of files available for transfer, and allow the client to find peer users known as *seeds* who may transfer the files.

Programmer Bram Cohen, a former University at Buffalo student,[4] designed the protocol in April 2001 and released the first available version on 2 July 2001,[5] and the most recent version in 2013.[6] BitTorrent clients are available for a variety of computing platforms and operating systems including an official client released by BitTorrent, Inc.

As of 2013, BitTorrent has 15–27 million concurrent users at any time.[7] As of January 2012, BitTorrent is utilized by 150 million active users. Based on this figure, the total number of monthly BitTorrent users may be estimated to more than a quarter of a billion.[8]

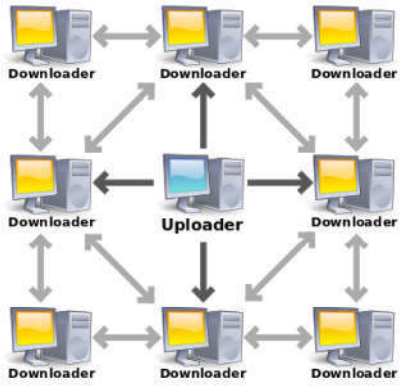
Contents



Animation of protocol use: The colored dots beneath each computer in the animation represent different parts of the file being shared. By the time a copy to a destination computer of each of those parts completes, a copy to another destination computer of that part (or other parts) is already taking place between users.

Description

The BitTorrent protocol can be used to reduce the server and network impact of distributing large files. Rather than downloading a file from a single source server, the BitTorrent protocol allows users to join a "swarm" of hosts to upload to/download from each other simultaneously. The protocol is an alternative to the older single source, multiple mirror sources technique for distributing data, and can work effectively over networks with lower bandwidth. Using the BitTorrent protocol, several basic computers, such as home computers, can replace large servers while efficiently distributing files to many recipients. This lower bandwidth usage also helps prevent large spikes in internet traffic in a given area, keeping internet speeds higher for all users in general, regardless of whether or not they use the BitTorrent protocol. A user who wants to upload a file first creates a small *torrent* descriptor file that they distribute by conventional means (web, email, etc.). They then make the file itself available through a BitTorrent node acting as a *seed*. Those with the torrent descriptor file can give it to their own BitTorrent nodes, which—acting as *peers* or *leechers*—download it by connecting to the seed and/or other peers (see diagram on the right).



The middle computer is acting as a "seed" to provide a file to the other computers which act as peers.

The file being distributed is divided into segments called *pieces*. As each peer receives a new piece of the file, it becomes a source (of that piece) for other peers, relieving the original seed from having to send that piece to every computer or user wishing a copy. With BitTorrent, the task of distributing the file is shared by those who want it; it is entirely possible for the seed to send only a single copy of the file itself and eventually distribute to an unlimited number of peers. Each piece is protected by a cryptographic hash contained in the torrent descriptor.[6] This ensures that any modification of the piece can be reliably detected, and thus prevents both accidental and malicious modifications of any of the pieces received at other nodes. If a node starts with an authentic copy of the torrent descriptor, it can verify the authenticity of the entire file it receives.

Pieces are typically downloaded non-sequentially and are rearranged into the correct order by the BitTorrent client, which monitors which pieces it needs, and which pieces it has and can upload to other peers. Pieces are of the same size throughout a single download (for example a 10 MB file may be transmitted as ten 1 MB pieces or as forty 256 KB pieces). Due to the nature of this approach, the download of any file can be halted at any time and be resumed at a later date, without the loss of previously downloaded information, which in turn makes BitTorrent particularly useful in the transfer of larger files. This also enables the client to seek out readily available pieces and download them immediately, rather than halting the download and waiting for the next (and possibly unavailable) piece in line, which typically reduces the overall time of the download. Once a peer has downloaded a file completely, it becomes an additional seed. This eventual transition from peers to seeders determines the overall "health" of the file (as determined by the number of times a file is available in its complete form).

The distributed nature of BitTorrent can lead to a flood-like spreading of a file throughout many peer computer nodes. As more peers join the swarm, the likelihood of a completely successful download by any particular node increases. Relative to traditional Internet distribution schemes, this permits a significant reduction in the original distributor's hardware and bandwidth resource costs. Distributed downloading protocols in general provide redundancy against system problems, reduce dependence on the original distributor[9] and provide sources for the file which are generally transient and therefore harder to trace by those who would block distribution compared to the situation provided by limiting availability of the file to a fixed host machine (or even several).

One such example of BitTorrent being used to reduce the distribution cost of file transmission is in the BOINC client-server system. If a BOINC distributed computing application needs to be updated (or merely sent to a user), it can do so with little impact on the BOINC server.[10]

Operation

A BitTorrent client is any program that implements the BitTorrent protocol. Each client is capable of preparing, requesting, and transmitting any type of computer file over a network, using the protocol. A peer is any computer running an instance of a client. To share a file or group of files, a peer first creates a small file called a "torrent" (e.g. MyFile.torrent). This file contains metadata about the files to be shared and about the tracker, the computer that coordinates the file distribution. Peers that want to download the file must first obtain a torrent file for it and connect to the specified tracker, which tells them from which other peers to download the pieces of the file.

Though both ultimately transfer files over a network, a BitTorrent download differs from a classic download (as is typical with an HTTP or FTP request, for example) in several fundamental ways:

- BitTorrent makes many small data requests over different IP connections to different machines, while classic downloading is typically made via a single TCP connection to a single machine.
- BitTorrent downloads in a random or in a "rarest-first"[11] approach that ensures high availability, while classic downloads are sequential.

Taken together, these differences allow BitTorrent to achieve much lower cost to the content provider, much higher redundancy, and much greater resistance to abuse or to "flash crowds" than regular server software. However, this protection, theoretically, comes at a cost: downloads can take time to rise to full speed because it may take time for enough peer connections to be established, and it may take time for a node to receive sufficient data to become an effective uploader. This contrasts with regular downloads (such as from an HTTP server, for example) that, while more vulnerable to overload and abuse, rise to full speed very quickly and maintain this speed throughout. In general, BitTorrent's non-contiguous download methods have prevented it from supporting progressive download or "streaming playback". However, comments made by Bram Cohen in January 2007[12] suggest that streaming torrent downloads will soon be commonplace and ad supported streaming[13] appears to be the result of those comments. In January 2011 Cohen demonstrated an early version of BitTorrent streaming, saying the feature was projected to be available by summer 2011.[14] As of 2013, this new BitTorrent streaming protocol is available for beta testing.[15]

Creating and publishing torrents

The peer distributing a data file treats the file as a number of identically sized pieces, usually with byte sizes of a power of 2, and typically between 32 kB and 16 MB each. The peer creates a hash for each piece, using the SHA-1 hash function, and records it in the torrent file. Pieces with sizes greater than 512 kB will reduce the size of a torrent file for a very large payload, but is claimed to reduce the efficiency of the protocol.[16] When another peer later receives a particular piece, the hash of the piece is compared to the recorded hash to test that the piece is error-free.[6] Peers that provide a complete file are called seeders, and the peer providing the initial copy is called the initial seeder. The exact information contained in the torrent file depends on the version of the BitTorrent protocol. By convention, the name of a torrent file has the suffix `.torrent`. Torrent files have an "announce" section, which specifies the URL of the tracker, and an "info" section, containing (suggested) names for the files, their lengths, the piece length used, and a SHA-1 hash code for each piece, all of which are used by clients to verify the integrity of the data they receive. Though SHA-1 has shown signs of cryptographic weakness, Bram Cohen did not initially consider the risk big enough for a backward incompatible change to, for example, SHA-3. BitTorrent is now preparing to move to SHA-256.

Torrent files are typically published on websites or elsewhere, and registered with at least one tracker. The tracker maintains lists of the clients currently participating in the torrent.[6] Alternatively, in a *trackerless system* (decentralized tracking) every peer acts as a tracker. Azureus was the first[17] BitTorrent client to implement such a system through the distributed hash table (DHT) method. An alternative and incompatible DHT system, known as Mainline DHT, was released in the Mainline BitTorrent client three weeks later (though it had been in development since 2002)[17] and subsequently adopted by the µTorrent, Transmission, rTorrent, KTorrent, BitComet, and Deluge clients.

After the DHT was adopted, a "private" flag – analogous to the broadcast flag – was unofficially introduced, telling clients to restrict the use of decentralized tracking regardless of the user's desires.[18] The flag is intentionally placed in the info section of the torrent so that it cannot be disabled or removed without changing the identity of the torrent. The purpose of the flag is to prevent torrents from being shared with clients that do not have access to the tracker. The flag was requested for inclusion in the official specification in August 2008, but has not been accepted yet.[19] Clients that have ignored the private flag were banned by many trackers, discouraging the practice.[20]

Downloading torrents and sharing files

Users find a torrent of interest, by browsing the web or by other means, download it, and open it with a BitTorrent client. The client connects to the tracker(s) specified in the torrent file, from which it receives a list of peers currently transferring pieces of the file(s) specified in the torrent. The client connects to those peers to obtain the various pieces. If the swarm contains only the initial seeder, the client connects directly to it and begins to request pieces. Clients incorporate mechanisms to optimize their download and upload rates; for example they download pieces in a random order to increase the opportunity to exchange data, which is only possible if two peers have different pieces of the file.

The effectiveness of this data exchange depends largely on the policies that clients use to determine to whom to send data. Clients may prefer to send data to peers that send data back to them (a "tit for tat" exchange scheme), which encourages fair trading. But strict policies often result in suboptimal situations, such as when newly joined peers are unable to receive any data because they don't have any pieces yet to trade themselves or when two peers with a good connection between them do not exchange data simply because neither of them takes the initiative. To counter these effects, the official BitTorrent client program uses a mechanism called "optimistic unchoking", whereby the client reserves a portion of its available bandwidth for sending pieces to random peers (not necessarily known good partners, so called preferred peers) in hopes of discovering even better partners and to ensure that newcomers get a chance to join the swarm.[21]

Although "swarming" scales well to tolerate "flash crowds" for popular content, it is less useful for unpopular or niche market content. Peers arriving after the initial rush might find the content unavailable and need to wait for the arrival of a "seed" in order to complete their downloads. The seed arrival, in turn, may take long to happen (this is termed the "seeder promotion problem"). Since maintaining seeds for unpopular content entails high bandwidth and administrative costs, this runs counter to the goals of publishers that value BitTorrent as a cheap alternative to a client-server approach. This occurs on a huge scale; measurements have shown that 38% of all new torrents become unavailable within the first month.[22] A strategy adopted by many publishers which significantly increases availability of unpopular content consists of bundling multiple files in a single swarm.[23] More sophisticated solutions have also been proposed; generally, these use cross-torrent mechanisms through which multiple torrents can cooperate to better share content.[24]

Concerns

BitTorrent does not, on its own, offer its users anonymity nor security. It is possible to obtain the IP addresses of all current and possibly previous participants in a swarm from the tracker. This may expose users with insecure systems to attacks.[21] It may also, in rare cases, expose users to the risk of being sued, if they are distributing files without permission from the copyright holder(s). However, there are ways to promote anonymity; for example, the OneSwarm project layers privacy-preserving sharing mechanisms on top of the original BitTorrent protocol. A moderate degree of anonymity, enough to keep ISPs from giving the user trouble at least, can be achieved with seedboxes which download the torrent files first to the companies' servers, followed by a direct download to the user.[25][26] Torrents can be downloaded with a high degree of anonymity by using services such as i2p. Tor does not provide anonymity on BitTorrent,[27] and its use is also discouraged (by blocking this type of connections) for performance reasons.[28] Unlike Tor, i2p is designed to work with BitTorrent[29] However, with i2p, torrents can only be downloaded from within the i2p network. This can be useful for users trying to avoid copyright complaints from their ISPs, maintaining privacy, or avoiding censorship.

Private trackers offer users a greater degree of privacy, compared to public trackers, but have the downside of a single centralized point of failure.

Bridging between i2p and the clearnet

Vuze is the only client that makes clearnet torrents available on i2p and vice versa. It has a plugin that connects to the i2p network. If the user adds a torrent from i2p or clearnet, it will be seeded on both i2p and the clearnet. For this reason, torrents previously published only on i2p are made available to the entire Internet, and users of i2p can download any torrent on the Internet while maintaining the anonymity of i2p.[30][31]

Adoption

A growing number of individuals and organizations are using BitTorrent to distribute their own or licensed works (e.g. indie bands distributing digital files of their new songs). Independent adopters report that without using BitTorrent technology, and its dramatically reduced demands on their private networking hardware and bandwidth, they could not afford to distribute their files.[32]

Some uses of BitTorrent for file sharing may violate laws in some jurisdictions (see legal issues section).

Film, video, and music

- BitTorrent Inc. has obtained a number of licenses from Hollywood studios for distributing popular content from their websites.
- Sub Pop Records releases tracks and videos via BitTorrent Inc.[33] to distribute its 1000+ albums. Babyshambles and The Libertines (both bands associated with Pete Doherty) have extensively used

torrents to distribute hundreds of demos and live videos. US industrial rock band Nine Inch Nails frequently distributes albums via BitTorrent.

- Podcasting software is starting to integrate BitTorrent to help podcasters deal with the download demands of their MP3 "radio" programs. Specifically, Juice and Miro (formerly known as Democracy Player) support automatic processing of .torrent files from RSS feeds. Similarly, some BitTorrent clients, such as µTorrent, are able to process web feeds and automatically download content found within them.
- DGM Live purchases are provided via BitTorrent.[34]
- VODO, a service which distributes "free-to-share" movies and TV shows via BitTorrent.[35][36][37]

Broadcasters

- In 2008, the CBC became the first public broadcaster in North America to make a full show (*Canada's Next Great Prime Minister*) available for download using BitTorrent.[38]
- The Norwegian Broadcasting Corporation (NRK) has since March 2008 experimented with bittorrent distribution, available online.[39] Only selected works in which NRK owns all royalties are published. Responses have been very positive, and NRK is planning to offer more content.
- The Dutch VPRO broadcasting organization released four documentaries in 2009 and 2010 under a Creative Commons license using the content distribution feature of the Mininova tracker.[40][41][42]

Personal works

- The Amazon S3 "Simple Storage Service" is a scalable Internet-based storage service with a simple web service interface, equipped with built-in BitTorrent support.[43]
- Blog Torrent offers a simplified BitTorrent tracker to enable bloggers and non-technical users to host a tracker on their site. Blog Torrent also allows visitors to download a "stub" loader, which acts as a BitTorrent client to download the desired file, allowing users without BitTorrent software to use the protocol.[44] This is similar to the concept of a self-extracting archive.

Software

- Blizzard Entertainment uses BitTorrent (via a proprietary client called the "Blizzard Downloader", associated with the Blizzard "BattleNet" network) to distribute content and patches for Diablo III, StarCraft II and World of Warcraft, including the games themselves.[45]
- Wargaming uses BitTorrent in their popular titles World of Tanks, World of Warships and World of Warplanes to distribute game updates.[46]
- CCP Games, maker of the space Simulation MMORPG Eve Online, has announced that a new launcher will be released that is based on BitTorrent.[47][48]
- Many software games, especially those whose large size makes them difficult to host due to bandwidth limits, extremely frequent downloads, and unpredictable changes in network traffic, will distribute instead a specialized, stripped down bittorrent client with enough functionality to download the game from the other running clients and the primary server (which is maintained in case not enough peers are available).
- Many major open source and free software projects encourage BitTorrent as well as conventional downloads of their products (via HTTP, FTP etc.) to increase availability and to reduce load on their own servers, especially when dealing with larger files.[49]

Government

- The UK government used BitTorrent to distribute details about how the tax money of UK citizens was spent.[50][51]

Education

- Florida State University uses BitTorrent to distribute large scientific data sets to its researchers.[52]
- Many universities that have BOINC distributed computing projects have used the BitTorrent functionality of the client-server system to reduce the bandwidth costs of distributing the client-side applications used to process the scientific data.
- The developing Human Connectome Project uses BitTorrent to share their open dataset (http://www.developingconnectome.org/project/data-release-user-guide/).[53]

Others

- Facebook uses BitTorrent to distribute updates to Facebook servers.[54]
- Twitter uses BitTorrent to distribute updates to Twitter servers.[55][56]
- The Internet Archive added BitTorrent to its file download options for over 1.3 million existing files, and all newly uploaded files, in August 2012.[57][58] This method is the fastest means of downloading media from the Archive.[57][59]

As of 2011, BitTorrent had 100 million users and a greater share of network bandwidth than Netflix and Hulu combined.[60][61] In early 2015, AT&T estimates that BitTorrent represents 20% of all broadband traffic.[62]

Routers that use network address translation (NAT) must maintain tables of source and destination IP addresses and ports. Typical home routers are limited to about 2000 table entries while some more expensive routers have larger table capacities. BitTorrent frequently contacts 20–30 servers per second, rapidly filling the NAT tables. This is a known cause of some home routers ceasing to work correctly.[63][64]

Indexing

The BitTorrent protocol provides no way to index torrent files. As a result, a comparatively small number of websites have hosted a large majority of torrents, many linking to copyrighted works without the authorization of copyright holders, rendering those sites especially vulnerable to lawsuits.[65] A BitTorrent index is a "list of .torrent files, which typically includes descriptions" and information about the torrent's content.[66] Several types of websites support the discovery and distribution of data on the BitTorrent network. Public torrent-hosting sites such as The Pirate Bay allow users to search and download from their collection of torrent files. Users can typically also upload torrent files for content they wish to distribute. Often, these sites also run BitTorrent trackers for their hosted torrent files, but these two functions are not mutually dependent: a torrent file could be hosted on one site and tracked by another unrelated site. Private host/tracker sites operate like public ones except that they may restrict access to registered users and may also keep track of the amount of data each user uploads and downloads, in an attempt to reduce "leeching".

Web search engines allow the discovery of torrent files that are hosted and tracked on other sites; examples include Mininova, BTDigg, BTJunkie, Torrentz, Torrentus, The Pirate Bay and isoHunt. These sites allow the user to ask for content meeting specific criteria (such as containing a given word or phrase) and retrieve a list of links to torrent files matching those criteria. This list can often be sorted with respect to several criteria, relevance (seeders-leechers ratio) being one of the most popular and useful (due to the way the protocol behaves, the download bandwidth achievable is very sensitive to this value). Bram Cohen launched a BitTorrent search engine on www.bittorrent.com/search (http://www.bittorrent.com/search) that co-mingles licensed content with search results.[67] Metasearch engines allow one to search several BitTorrent indices and search engines at once. DHT search engines monitors the DHT network and indexes torrents via metadata exchange from peers. In the 2010s, some P2P, decentralized alternatives to Torrent search engines have emerged, see decentralized keyword search below.

Technologies built on BitTorrent

The BitTorrent protocol is still under development and may therefore still acquire new features and other enhancements such as improved efficiency.

Distributed trackers

On 2 May 2005, Azureus 2.3.0.0 (now known as Vuze) was released,[68] introducing support for "trackerless" torrents through a system called the "distributed database." This system is a Distributed hash table implementation which allows the client to use torrents that do not have a working BitTorrent tracker. The following month, BitTorrent, Inc. released version 4.2.0 of the Mainline BitTorrent client, which supported an alternative DHT implementation (popularly known as "Mainline DHT", outlined in a draft on their website) that is incompatible with that of Azureus. Recent measurement shows users of Mainline DHT is from 10 million to 25 million, with a daily churn of at least 10 million.[69] Mainline DHT is arguably the largest realistic DHT in the world.

Current versions of the official BitTorrent client, µTorrent, BitComet, Transmission and BitSpirit all share compatibility with Mainline DHT. Both DHT implementations are based on Kademlia.[70] As of version 3.0.5.0, Azureus also supports Mainline DHT in addition to its own distributed database through use of an optional application plugin.[71] This potentially allows the Azureus/Vuze client to reach a bigger swarm.

Another idea that has surfaced in Vuze is that of *virtual torrents*. This idea is based on the distributed tracker approach and is used to describe some web resource. Currently, it is used for instant messaging. It is implemented using a special messaging protocol and requires an appropriate plugin. Anatomic P2P is another approach, which uses a decentralized network of nodes that route traffic to dynamic trackers. Most BitTorrent clients also use Peer exchange (PEX) to gather peers in addition to trackers and DHT. Peer exchange checks with known peers to see if they know of any other peers. With the 3.0.5.0 release of Vuze, all major BitTorrent clients now have compatible peer exchange.

Web seeding

Web "seeding" was implemented in 2006 as the ability of BitTorrent clients to download torrent pieces from an HTTP source in addition to the "swarm". The advantage of this feature is that a website may distribute a torrent for a particular file or batch of files and make those files available for download from that same web server; this can simplify long-term seeding and load balancing through the use of existing, cheap, web hosting setups. In theory, this would make using BitTorrent almost as easy for a web publisher as creating a direct HTTP download. In addition, it would allow the "web seed" to be disabled if the swarm becomes too popular while still allowing the file to be readily available. This feature has two distinct specifications, both of which are supported by Libtorrent and the 26+ clients that use it.

Hash web seeding

The first was created by John "TheSHADoW" Hoffman, who created BitTornado.[72][73] This first specification requires running a web service that serves content by info-hash and piece number, rather than filename.

HTTP web seeding

The other specification is created by GetRight authors and can rely on a basic HTTP download space (using byte serving).[74][75]

Other

In September 2010, a new service named Burnbit was launched which generates a torrent from any URL using webseeding.[76] There are server-side solutions that provide initial seeding of the file from the webserver via standard BitTorrent protocol and when the number of external seeders reach a limit, they stop serving the file from the original

source.[77]

RSS feeds

A technique called broadcatching combines RSS feeds with the BitTorrent protocol to create a content delivery system, further simplifying and automating content distribution. Steve Gillmor explained the concept in a column for Ziff-Davis in December 2003.[78] The discussion spread quickly among bloggers (Ernest Miller,[79] Chris Pirillo, etc.). In an article entitled *Broadcatching with BitTorrent*, Scott Raymond explained:

> I want RSS feeds of BitTorrent files. A script would periodically check the feed for new items, and use them to start the download. Then, I could find a trusted publisher of an Alias RSS feed, and "subscribe" to all new episodes of the show, which would then start downloading automatically – like the "season pass" feature of the TiVo.
>
> — Scott Raymond, scottraymond.net[80]

The RSS feed will track the content, while BitTorrent ensures content integrity with cryptographic hashing of all data, so feed subscribers will receive uncorrupted content. One of the first and popular software clients (free and open source) for *broadcatching* is Miro. Other free software clients such as PenguinTV and KatchTV are also now supporting broadcatching. The BitTorrent web-service MoveDigital added the ability to make torrents available to any web application capable of parsing XML through its standard REST-based interface in 2006,[81] though this has since been discontinued. Additionally, Torrenthut is developing a similar torrent API that will provide the same features, and help bring the torrent community to Web 2.0 standards. Alongside this release is a first PHP application built using the API called PEP, which will parse any Really Simple Syndication (RSS 2.0) feed and automatically create and seed a torrent for each enclosure found in that feed.[82]

Throttling and encryption

Since BitTorrent makes up a large proportion of total traffic, some ISPs have chosen to "throttle" (slow down) BitTorrent transfers. For this reason, methods have been developed to disguise BitTorrent traffic in an attempt to thwart these efforts.[83] Protocol header encrypt (PHE) and Message stream encryption/Protocol encryption (MSE/PE) are features of some BitTorrent clients that attempt to make BitTorrent hard to detect and throttle. As of November 2015, Vuze, Bitcomet, KTorrent, Transmission, Deluge, µTorrent, MooPolice, Halite, qBittorrent, rTorrent, and the latest official BitTorrent client (v6) support MSE/PE encryption. In September 2006 it was reported that some software could detect and throttle BitTorrent traffic masquerading as HTTP traffic.[84]

Reports in August 2007 indicated that Comcast was preventing BitTorrent seeding by monitoring and interfering with the communication between peers. Protection against these efforts is provided by proxying the client-tracker traffic via an encrypted tunnel to a point outside of the Comcast network.[85] Comcast has more recently called a "truce" with BitTorrent, Inc. with the intention of shaping traffic in a protocol-agnostic manner.[86] Questions about the ethics and legality of Comcast's behavior have led to renewed debate about net neutrality in the United States.[87] In general, although encryption can make it difficult to determine *what* is being shared, BitTorrent is vulnerable to traffic analysis. Thus, even with MSE/PE, it may be possible for an ISP to recognize BitTorrent and also to determine that a system is no longer downloading but only uploading data, and terminate its connection by injecting TCP RST (reset flag) packets.

Multitracker

Another unofficial feature is an extension to the BitTorrent metadata format proposed by John Hoffman[88] and implemented by several indexing websites. It allows the use of multiple trackers per file, so if one tracker fails, others can continue to support file transfer. It is implemented in several clients, such as BitComet, BitTornado, BitTorrent, KTorrent, Transmission, Deluge, µTorrent, rtorrent, Vuze, and Frostwire. Trackers are placed in groups, or tiers, with a tracker randomly chosen from the top tier and tried, moving to the next tier if all the trackers in the top tier fail.

Torrents with multiple trackers can decrease the time it takes to download a file, but also have a few consequences:

- Poorly implemented[89] clients may contact multiple trackers, leading to more overhead-traffic.
- Torrents from closed trackers suddenly become downloadable by non-members, as they can connect to a seed via an open tracker.

Decentralized keyword search

Even with distributed trackers, a third party is still required to find a specific torrent. This is usually done in the form of a hyperlink from the website of the content owner or through indexing websites like isoHunt, Torrentz, BTDigg, Torrentus or The Pirate Bay. The Tribler BitTorrent client is the first to incorporate decentralized search capabilities.

With Tribler, users can find .torrent files that are hosted among other peers, instead of on a centralized index sites. It adds such an ability to the BitTorrent protocol using a gossip protocol, somewhat similar to the eXeem network which was shut down in 2005. The software includes the ability to recommend content as well. After a dozen downloads the Tribler software can roughly estimate the download taste of the user and recommend additional content.[90]

In May 2007, researches at Cornell University published a paper proposing a new approach to searching a peer-to-peer network for inexact strings,[91] which could replace the functionality of a central indexing site. A year later, the same team implemented the system as a plugin for Vuze called Cubit[92] and published a follow-up paper reporting its success.[93]

A somewhat similar facility but with a slightly different approach is provided by the BitComet client through its "Torrent Exchange"[94] feature. Whenever two peers using BitComet (with Torrent Exchange enabled) connect to each other they exchange lists of all the torrents (name and info-hash) they have in the Torrent Share storage (torrent files which were previously downloaded and for which the user chose to enable sharing by Torrent Exchange). Thus each client builds up a list of all the torrents shared by the peers it connected to in the current session (or it can even maintain the list between sessions if instructed). At any time the user can search into that Torrent Collection list for a certain torrent and sort the list by categories. When the user chooses to download a torrent from that list, the .torrent file is automatically searched for (by info-hash value) in the DHT Network and when found it is downloaded by the querying client which can after that create and initiate a downloading task.

Implementations

The BitTorrent specification is free to use and many clients are open source, so BitTorrent clients have been created for all common operating systems using a variety of programming languages. The official BitTorrent client, µTorrent, qBittorrent, Transmission, Vuze, and BitComet are some of the most popular clients.[95][96][97][98]

Some BitTorrent implementations such as MLDonkey and Torrentflux are designed to run as servers. For example, this can be used to centralize file sharing on a single dedicated server which users share access to on the network.[99] Server-oriented BitTorrent implementations can also be hosted by hosting providers at co-located facilities with high bandwidth Internet connectivity (e.g., a datacenter) which can provide dramatic speed benefits over using BitTorrent from a regular home broadband connection. Services such as ImageShack can download files on BitTorrent for the user, allowing them to download the entire file by HTTP once it is finished. The Opera web browser supports BitTorrent,[100] as does Wyzo. BitLet allows users to download Torrents directly from their browser using a Java

applet. An increasing number of hardware devices are being made to support BitTorrent. These include routers and NAS devices containing BitTorrent-capable firmware like OpenWrt. Proprietary versions of the protocol which implement DRM, encryption, and authentication are found within managed clients such as Pando.

Development

An unimplemented (as of February 2008) unofficial feature is Similarity Enhanced Transfer (SET), a technique for improving the speed at which peer-to-peer file sharing and content distribution systems can share data. SET, proposed by researchers Pucha, Andersen, and Kaminsky, works by spotting chunks of identical data in files that are an exact or near match to the one needed and transferring these data to the client if the "exact" data are not present. Their experiments suggested that SET will help greatly with less popular files, but not as much for popular data, where many peers are already downloading it.[101] Andersen believes that this technique could be immediately used by developers with the BitTorrent file sharing system.[102]

As of December 2008, BitTorrent, Inc. is working with Oversi on new Policy Discover Protocols that query the ISP for capabilities and network architecture information. Oversi's ISP hosted NetEnhancer box is designed to "improve peer selection" by helping peers find local nodes, improving download speeds while reducing the loads into and out of the ISP's network.[103]

Legal issues

Although the protocol itself is legal,[104] problems stem from using the protocol to traffic copyright infringing works. There has been much controversy over the use of BitTorrent trackers. BitTorrent metafiles themselves do not store file contents. Whether the publishers of BitTorrent metafiles violate copyrights by linking to copyrighted works without the authorization of copyright holders is controversial. Various jurisdictions have pursued legal action against websites that host BitTorrent trackers. High-profile examples include the closing of Suprnova.org, TorrentSpy, LokiTorrent, BTJunkie, Mininova, Demonoid and Oink's Pink Palace. The Pirate Bay torrent website, formed by a Swedish group, is noted for the "legal" section of its website in which letters and replies on the subject of alleged copyright infringements are publicly displayed. On 31 May 2006, The Pirate Bay's servers in Sweden were raided by Swedish police on allegations by the MPAA of copyright infringement;[105] however, the tracker was up and running again three days later. In the study used to value NBC Universal in its merger with Comcast, Envisional examined the 10,000 torrent swarms managed by PublicBT which had the most active downloaders. After excluding pornographic and unidentifiable content, it was found that only one swarm offered legitimate content.[106]

In the United States, more than 200,000 people have been sued for filesharing on BitTorrent since 2010.[107] On 30 April 2012, the UK High Court ordered five ISPs to block BitTorrent search engine The Pirate Bay.[108]

Security problems

BitTorrent implementations often use µTP for their communication. To achieve high bandwidths, the underlying protocol used is UDP, which allows spoofing of source addresses of internet traffic. This can be used for Denial-of-service attacks, where users running BitTorrent clients act as amplifiers for an attack at another service.[109]

Challenges

"Leeches", are those users who download more than they share. As BitTorrent is a collaborative distributed platform, there is a section of the community that wants solutions to punish and discourage such behaviour.[110]

Malware

Several studies on BitTorrent have indicated that there exist files, containing malware, available for download via BitTorrent. In particular, one small sample[111] indicated that 18% of all executable programs available for download contained malware. Another study[112] claims that as much as 14.5% of BitTorrent downloads contain zero-day malware, and that BitTorrent was used as the distribution mechanism for 47% of all zero-day malware they have found.

BitErrant attack

Due to SHA1 collisions, an attacker can alter the execution path of the executable by serving altered chunks when the victim is downloading the executable using the BitTorrent protocol.[113]

Criticism of BitErrant attack

Despite the fact that a proof of concept exists, the attack may succeed in very limited cases: such as small chunk size (32kB). By selecting larger chunks (i.e. >256kB) the amount of resources required to find SHA1 collision is tremendous, which makes the attack virtually impossible.

See also

- BitTorrent portal
- Anonymous P2P
- Napster
- Gnutella
- Anti-Counterfeiting Trade Agreement
- Bencode
- Cache Discovery Protocol
- Comparison of BitTorrent clients
- Comparison of BitTorrent sites
- Comparison of BitTorrent tracker software
- FastTrack
- Glossary of BitTorrent terms
- Magnet URI scheme
- µTP (Micro Transport Protocol)
- Peer-to-peer file sharing
- Segmented file transfer
- Simple file verification
- Super-seeding
- Torrent file
- Torrent poisoning
- VPN

References

1. Schulze, Hendrik; Klaus Mochalski (2009). "Internet Study 2008/2009" (https://www.webcitation.org/6OVSh9hZ0?url=http://www.ipoque.com/sites/default/files/mediafiles/documents/internet-study-2008-2009.pdf) (PDF). Leipzig, Germany: ipoque. Archived from the original (http://www.ipoque.com/sites/default/files/mediafiles/documents/internet-study-2008-2009.pdf) (PDF) on 1 April 2014. Retrieved 3 October 2011. "Peer-to-peer file sharing (P2P) still generates by far the most traffic in all monitored regions – ranging from 43% in Northern Africa to 70% Eastern Europe."

2. "Application Usage & Threat Report" (http://researchcenter.paloaltonetworks.com/app-usage-risk-report-visualization/). Palo Alto Networks. 2013. Archived (https://web.archive.org/web/20131031153132/http://researchcenter.paloaltonetworks.com/app-usage-risk-report-visualization/) from the original on 31 October 2013. Retrieved 7 April 2013.
3. Van der Sar, Ernesto (4 December 2009). "Thunder Blasts uTorrent's Market Share Away - TorrentFreak" (https://web.archive.org/web/20160220105711/https://torrentfreak.com/thunder-blasts-utorrents-market-share-away-091204/). *TorrentFreak*. Archived from the original (https://torrentfreak.com/thunder-blasts-utorrents-market-share-away-091204/) on 20 February 2016. Retrieved 18 June 2018.
4. "UB Engineering Tweeter" (https://twitter.com/UBengineering/status/357131210503356419). University at Buffalo's School of Engineering and Applied Sciences. Archived (https://web.archive.org/web/20131111070426/https://twitter.com/UBengineering/status/357131210503356419) from the original on 11 November 2013.
5. Cohen, Bram (2 July 2001). "BitTorrent – a new P2P app" (http://finance.groups.yahoo.com/group/decentralization/message/3160). Yahoo eGroups. Archived (https://www.webcitation.org/5zslWzAdB?url=http://finance.groups.yahoo.com/group/decentralization/message/3160) from the original on 2 July 2011. Retrieved 15 April 2007.
6. Cohen, Bram (October 2002). "BitTorrent Protocol 1.0" (http://www.bittorrent.org/beps/bep_0003.html). BitTorrent.org. Archived (https://web.archive.org/web/20140208002821/http://bittorrent.org/beps/bep_0003.html) from the original on 8 February 2014. Retrieved 19 April 2013.
7. Wang, Liang; Kangasharju, J. (1 September 2013). "Measuring large-scale distributed systems: Case of Bit *Torrent* Mainline DHT". *IEEE P2P 2013 Proceedings* (https://www.cl.cam.ac.uk/~lw525/MLDHT/). pp. 1–10. doi:10.1109/P2P.2013.6688697 (https://doi.org/10.1109%2FP2P.2013.6688697). ISBN 978-1-4799-0515-7. Retrieved 7 January 2016.
8. "BitTorrent and µTorrent Software Surpass 150 Million User Milestone" (https://web.archive.org/web/20140326102305/http://www.bittorrent.com/intl/es/company/about/ces_2012_150m_users). Bittorrent.com. 9 January 2012. Archived from the original (http://www.bittorrent.com/intl/es/company/about/ces_2012_150m_users) on 26 March 2014. Retrieved 9 July 2012.
9. Menasche, Daniel S.; Rocha, Antonio A. A.; de Souza e Silva, Edmundo A.; Leao, Rosa M.; Towsley, Don; Venkataramani, Arun (2010). "Estimating Self-Sustainability in Peer-to-Peer Swarming Systems". arXiv:1004.0395 (https://arxiv.org/abs/1004.0395) [cs.NI (https://arxiv.org/archive/cs.NI)]. by D. Menasche, A. Rocha, E. de Souza e Silva, R. M. Leao, D. Towsley, A. Venkataramani.
10. Costa, Fernando; Silva, Luis; Fedak, Gilles; Kelley, Ian (2008). "Optimizing the data distribution layer of BOINC with Bit *Torrent*". *2008 IEEE International Symposium on Parallel and Distributed Processing* (https://www.academia.edu/583565/Optimizing_the_data_distribution_layer_of_boinc_with_bittorrent._Parallel_and_Distributed_Processing_2008._IPDPS_2008) (PDF). *IEEE International Symposium on Parallel and Distributed Processing, 2008. IPDPS 2008*. IEEE. p. 1. doi:10.1109/IPDPS.2008.4536446 (https://doi.org/10.1109%2FIPDPS.2008.4536446). ISBN 978-1-4244-1693-6. Retrieved 7 April 2013.(registration required)
11. Urvoy-Keller (December 2006). "Rarest First and Choke Algorithms Are Enough" (http://conferences.sigcomm.org/imc/2006/papers/p20-legout.pdf) (PDF). SIGCOMM. Retrieved 9 March 2012.
12. Ernesto (17 January 2007). "Interview with Bram Cohen, the inventor of BitTorrent" (http://torrentfreak.com/interview-with-bram-cohen-the-inventor-of-bittorrent). TorrentFreak. Archived (https://web.archive.org/web/20130715130429/http://torrentfreak.com/interview-with-bram-cohen-the-inventor-of-bittorrent/) from the original on 15 July 2013. Retrieved 7 April 2013.
13. Ernesto (18 December 2007). "BitTorrent Launches Ad Supported Streaming" (http://torrentfreak.com/bittorrent-launches-ad-supported-streaming-071218). TorrentFreak. Archived (https://web.archive.org/web/20130619184917/http://torrentfreak.com/bittorrent-launches-ad-supported-streaming-071218/) from the original on 19 June 2013. Retrieved 7 April 2013.
14. Cohen, Bram (January 2011). "BitTorrent Inventor Demos New P2P Live Streaming Protocol" (http://torrentfreak.com/bittorrent-p2p-live-streaming-110119/). torrentfreak.com. Archived (https://web.archive.org/web/20131027114200/http://torrentfreak.com/bittorrent-p2p-live-streaming-110119/) from the original on 27 October 2013. Retrieved 19 January 2011.
15. "BitTorrent Live" (https://web.archive.org/web/20131025133454/http://live.bittorrent.com/). Bittorrent.com. Archived from the original (http://live.bittorrent.com/) on 25 October 2013. Retrieved 4 March 2013.

16. "BitTorrent Specification" (http://wiki.theory.org/index.php/BitTorrentSpecification). Wiki.theory.org. Archived (https://web.archive.org/web/20130626195027/https://wiki.theory.org/index.php/BitTorrentSpecification) from the original on 26 June 2013. Retrieved 9 July 2012.
17. Jones, Ben (7 June 2015). "BitTorrent's DHT Turns 10 Years Old" (https://torrentfreak.com/bittorrents-dht-turns-10-years-old-150607/). *TorrentFreak*. Retrieved 5 July 2015.
18. "Unofficial BitTorrent Protocol Specification v1.0" (http://wiki.theory.org/BitTorrentSpecification#Info_Dictionary). Archived (https://web.archive.org/web/20061214094732/http://wiki.theory.org/BitTorrentSpecification) from the original on 14 December 2006. Retrieved 4 October 2009.
19. Harrison, David (3 August 2008). "Private Torrents" (http://bittorrent.org/beps/bep_0027.html). Bittorrent.org. Archived (https://web.archive.org/web/20130324181003/http://www.bittorrent.org/beps/bep_0027.html) from the original on 24 March 2013. Retrieved 4 October 2009.
20. "BitComet Banned From Growing Number of Private Trackers" (http://www.slyck.com/news.php?story=1021). Archived (https://web.archive.org/web/20140326093846/http://www.slyck.com/news.php?story=1021) from the original on 26 March 2014. Retrieved 4 October 2009.
21. Tamilmani, Karthik (25 October 2003). "Studying and enhancing the BitTorrent protocol" (https://web.archive.org/web/20041119150847/http://mnl.cs.stonybrook.edu/home/karthik/BitTorrent/Robustness_of_BT.doc). Stony Brook University. Archived from the original (http://mnl.cs.stonybrook.edu/home/karthik/BitTorrent/Robustness_of_BT.doc) (DOC) on 19 November 2004. Retrieved 6 May 2006.
22. Kaune, Sebastian; et al. (2009). "Unraveling BitTorrent's File Unavailability: Measurements and Analysis". arXiv:0912.0625 (https://arxiv.org/abs/0912.0625) [cs.NI (https://arxiv.org/archive/cs.NI)].
23. D. Menasche; et al. (December 1–4, 2009). *Content Availability and Bundling in Swarming Systems* (http://conferences.sigcomm.org/co-next/2009/papers/Menasche.pdf) (PDF). *CoNEXT'09*. Rome, Italy: ACM via sigcomm.org. ISBN 978-1-60558-636-6.
24. Kaune, Sebastian; et al. "The Seeder Promotion Problem: Measurements, Analysis and Solution Space" (http://www.eecs.qmul.ac.uk/~tysong/files/ICCCN09.pdf) (PDF). Queen Mary's University London. Retrieved 20 July 2017.
25. "This Website Could Be The Ultimate All-In-One Torrent Machine" (https://web.archive.org/web/20160408154942/http://gizmodo.com/this-website-could-be-the-ultimate-all-in-one-torrent-m-1677265492). 8 April 2016. Archived from the original (https://gizmodo.com/this-website-could-be-the-ultimate-all-in-one-torrent-m-1677265492) on 8 April 2016.
26. "Torrent From the Cloud With Seedr - TorrentFreak" (https://torrentfreak.com/torrent-from-the-cloud-with-seedr-160117/). 17 January 2016.
27. "Bittorrent over Tor isn't a good idea - The Tor Blog" (https://blog.torproject.org/blog/bittorrent-over-tor-isnt-good-idea).
28. Inc., The Tor Project,. "Tor Project: FAQ" (https://www.torproject.org/docs/faq.html.en#FileSharing).
29. "I2P Compared to Tor - I2P" (https://geti2p.net/en/comparison/tor).
30. "Vuze Speeds Up Torrent Downloads Through "Swarm Merging" - TorrentFreak" (https://torrentfreak.com/vuze-speeds-up-torrent-downloads-through-swarm-merging-150320/). 20 March 2015.
31. "I2PHelper HowTo - VuzeWiki" (https://wiki.vuze.com/w/I2PHelper_HowTo#Network_Mixing).
32. See, for example, "Why Bit Torrent" (http://tasvideos.org/WhyBitTorrent.html). Archived (https://web.archive.org/web/20130128060756/http://tasvideos.org/WhyBitTorrent.html) from the original on 28 January 2013.. tasvideos.org.
33. "Sub Pop page on BitTorrent.com" (https://web.archive.org/web/20070114140652/http://www.bittorrent.com/users/subpoprecords/). Archived from the original (http://www.bittorrent.com/users/subpoprecords/) on 14 January 2007. Retrieved 13 December 2006.
34. "DGMlive.com" (http://www.dgmlive.com/help.htm#whatisbittorrent). DGMlive.com. Archived (https://web.archive.org/web/20131111075532/https://www.dgmlive.com/help.htm) from the original on 11 November 2013. Retrieved 9 July 2012.
35. "VODO – About..." (http://vo.do/about). Retrieved 15 April 2012. (WebCite (https://www.webcitation.org/66wxu53jV?url=http://vo.do/about)).

36. Cory Doctorow (15 October 2009). "Vodo: a filesharing service for film-makers" (http://boingboing.net/2009/10/15/vodo-a-filesharing-s.html). Boing Boing. Happy Mutants LLC. Retrieved 15 April 2012. (WebCite (https://www.webcitation.org/66wy0PFq1?url=http://boingboing.net/2009/10/15/vodo-a-filesharing-s.html))
37. Ernesto. "Pioneer One, The BitTorrent Exclusive TV-Series Continues" (https://torrentfreak.com/pioneer-one-the-bittorrent-exclusive-tv-series-continues-101215/). TorrentFreak. Retrieved 15 April 2012. (WebCite (https://www.webcitation.org/66wyOrilB?url=https://torrentfreak.com/pioneer-one-the-bittorrent-exclusive-tv-series-continues-101215/))
38. "CBC to BitTorrent Canada's Next Great Prime Minister:" (https://web.archive.org/web/20100614220807/http://www.cbc.ca/nextprimeminister/blog/2008/03/canadas_next_great_prime_minis.html). *CBC News*. 19 March 2008. Archived from the original (http://www.cbc.ca/nextprimeminister/blog/2008/03/canadas_next_great_prime_minis.html) on 14 June 2010. Retrieved 19 March 2008.
39. "Bittorrent" (http://nrkbeta.no/bittorrent/) (in Norwegian). Nrkbeta.no. 2008. Archived (https://web.archive.org/web/20131024144950/http://nrkbeta.no/bittorrent/) from the original on 24 October 2013. Retrieved 7 April 2013.
40. "Torrents uploaded by EeuwvandeStad" (https://web.archive.org/web/20131104210531/http://www.mininova.org/user/EeuwvandeStad). MiniNova. 2009. Archived from the original (http://www.mininova.org/user/EeuwvandeStad) on 4 November 2013. Retrieved 7 April 2013.
41. Denters, M. (11 August 2010). "Tegenlicht – Download California Dreaming" (http://tegenlicht.vpro.nl/nieuws/2010/november/creative-commons.html). VPRO.nl. Archived (https://web.archive.org/web/20140326102335/http://tegenlicht.vpro.nl/nieuws/2010/november/creative-commons.html) from the original on 26 March 2014. Retrieved 7 April 2013.
42. Bol, M. (1 October 2009). "Tegenlicht – VPRO gemeengoed" (http://tegenlicht.vpro.nl/nieuws/2009/oktober/vpro-gemeengoed.html) (in Dutch). VPRO.nl. Archived (https://web.archive.org/web/20140326094708/http://tegenlicht.vpro.nl/nieuws/2009/oktober/vpro-gemeengoed.html) from the original on 26 March 2014. Retrieved 7 April 2013.
43. "Using BitTorrent with Amazon S3" (http://docs.aws.amazon.com/AmazonS3/latest/dev/S3Torrent.html). Archived (https://web.archive.org/web/20140326040056/http://docs.aws.amazon.com/AmazonS3/latest/dev/S3Torrent.html) from the original on 26 March 2014.
44. Rustad, Roger E. (26 August 2004). "Blog Torrent and Participatory Culture" (https://web.archive.org/web/20060612202944/http://grep.law.harvard.edu/article.pl?sid=04%2F08%2F26%2F0236209). Grep Law. Archived from the original (http://grep.law.harvard.edu/article.pl?sid=04/08/26/0236209) on 12 June 2006. Retrieved 9 May 2006.
45. "Blizzard Downloader" (http://www.wowpedia.org/Blizzard_Downloader). Curse Inc. 4 November 2010. Archived (https://web.archive.org/web/20140326100329/http://wowpedia.org/Blizzard_Downloader) from the original on 26 March 2014. Retrieved 4 November 2010.
46. "World of Tanks FAQ" (http://worldoftanks.eu/en/content/guide/general/frequently_asked_questions/). Wargaming. 15 December 2014.
47. MJ Guthrie (11 March 2013). "EVE Online reconfiguring launcher to use BitTorrent" (http://massively.joystiq.com/2013/03/11/eve-online-reconfiguring-launcher-to-use-bittorrent/). Massively.joystiq.com. Archived (https://web.archive.org/web/20140213023755/http://massively.joystiq.com/2013/03/11/eve-online-reconfiguring-launcher-to-use-bittorrent/) from the original on 13 February 2014. Retrieved 7 April 2013.
48. CCP Games (20 July 2010). "All quiet on the EVE Launcher front? – EVE Community" (http://community.eveonline.com/devblog.asp?a=blog&nbid=74573). Community.eveonline.com. Archived (https://web.archive.org/web/20130313171657/http://community.eveonline.com/devblog.asp?a=blog&nbid=74573) from the original on 13 March 2013. Retrieved 7 April 2013.
49. "Complete Download Options List – BitTorrent" (http://www.ubuntu.com/getubuntu/downloadmirrors#bt). Ubuntu.com. Archived (https://web.archive.org/web/20100424013939/http://www.ubuntu.com/getubuntu/downloadmirrors) from the original on 24 April 2010. Retrieved 7 May 2009.
50. HM Government (4 September 2012). "Combined Online Information System" (http://data.gov.uk/dataset/coins). *Data.Gov.Uk Beta*. Controller of Her Majesty's Stationery Office. Archived (https://web.archive.org/web/20140326093701/http://data.gov.uk/dataset/coins) from the original on 26 March 2014. Retrieved 7 September 2012.

51. Ernesto (4 June 2010). "UK Government Uses BitTorrent to Share Public Spending Data" (http://torrentfreak.com/uk-government-uses-bittorrent-to-share-public-spending-data-100604/). TorrentFreak. Archived (https://web.archive.org/web/20131027201557/http://torrentfreak.com/uk-government-uses-bittorrent-to-share-public-spending-data-100604/) from the original on 27 October 2013. Retrieved 7 September 2012.
52. "HPC Data Repository" (http://www.hpc.fsu.edu/index.php?option=com_wrapper&view=wrapper&Itemid=80). Florida State University. Retrieved 7 April 2013.
53. "Torrents Help Researchers Worldwide to Study Babies' Brains" (https://torrentfreak.com/torrents-help-researchers-worldwide-to-study-babies-brains-170603/). Torrent Freak. 3 June 2017. Retrieved 4 January 2018.
54. Ernesto (25 June 2010). "Facebook Uses BitTorrent, and They Love It" (http://torrentfreak.com/facebook-uses-bittorrent-and-they-love-it-100625/). *Torrent Freak*. Torrent Freak. Archived (https://web.archive.org/web/20140419233159/https://torrentfreak.com/facebook-uses-bittorrent-and-they-love-it-100625/) from the original on 19 April 2014. Retrieved 7 September 2012.
55. Ernesto (10 February 2010). "Twitter Uses BitTorrent For Server Deployment" (http://torrentfreak.com/twitter-uses-bittorrent-for-server-deployment-100210/). *Torrent Freak*. Torrent Freak. Archived (https://web.archive.org/web/20140326094120/http://torrentfreak.com/twitter-uses-bittorrent-for-server-deployment-100210/) from the original on 26 March 2014. Retrieved 7 September 2012.
56. Ernesto (16 July 2010). "BitTorrent Makes Twitter's Server Deployment 75x Faster" (http://torrentfreak.com/bittorrent-makes-twitters-server-deployment-75-faster-100716/). *Torrent Freak*. Torrent Freak. Archived (https://web.archive.org/web/20140326093158/http://torrentfreak.com/bittorrent-makes-twitters-server-deployment-75-faster-100716/) from the original on 26 March 2014. Retrieved 7 September 2012.
57. Ernesto (7 August 2012). "Internet Archive Starts Seeding 1,398,875 Torrents" (https://torrentfreak.com/internet-archive-starts-seeding-1398635-torrents-120807/). TorrentFreak. Archived (https://www.webcitation.org/69IfceFOU?url=https://torrentfreak.com/internet-archive-starts-seeding-1398635-torrents-120807/) from the original on 8 August 2012. Retrieved 7 August 2012.
58. "Hot List for bt1.us.archive.org (Updated August 7, 2012, 7:31 pm PDT)" (http://bt1.archive.org/hotlist.php).. Archive.org.
59. "Welcome to Archive torrents" (https://archive.org/details/bittorrent).. Archive.org. 2012.
60. Carr, Austin (4 January 2011). "BitTorrent Has More Users Than Netflix and Hulu Combined—and Doubled" (http://www.fastcompany.com/1714001/bittorrent-swells-to-100-million-users). fastcompany.com. Archived (https://web.archive.org/web/20110110112552/http://www.fastcompany.com//1714001//bittorrent-swells-to-100-million-users) from the original on 10 January 2011. Retrieved 9 July 2012.
61. Hartley, Matt (1 July 2011). "BitTorrent turns ten" (http://business.financialpost.com/2011/07/01/bittorrent-turns-ten/). Financialpost.com. Archived (https://web.archive.org/web/20131104213456/http://business.financialpost.com/2011/07/01/bittorrent-turns-ten/) from the original on 4 November 2013. Retrieved 9 July 2012.
62. "AT&T patents system to 'fast-lane' BitTorrent traffic" (http://thestack.com/atandt-patents-system-speed-up-bittorrent-traffic-190215). Thestack.com. 8 May 2006. Retrieved 5 March 2015.
63. "FAQ:Modems/routers that are known to have problems with P2P apps" (https://web.archive.org/web/20080913085527/http://www.utorrent.com/faq.php#Modems_routers_that_are_known_to_have_problems_with_P2P). uTorrent.com. Archived from the original (http://www.utorrent.com/faq.php#Modems_routers_that_are_known_to_have_problems_with_P2P) on 13 September 2008. Retrieved 7 April 2013.
64. Halkes, Gertjan; Pouwelse, Johan (2011). Jordi Domingo-Pascual et al., eds. *UDP NAT and Firewall Puncturing in the Wild* (https://books.google.com/books?id=j6HPyxvPFcsC&pg=PA7#v=onepage&f=false). *NETWORKING 2011:10th International IFIP TC 6 Networking Conference, Valencia, Spain, May 9–13, 2011, Proceedings*. Springer. p. 7. ISBN 9783642207976. Retrieved 7 April 2013.
65. Ernesto (12 July 2009). "PublicBT Tracker Set To Patch BitTorrent' Achilles' Heel" (http://torrentfreak.com/publicbt-tracker-set-to-patch-bittorrents-achilles-heel-090712/). Torrentfreak. Archived (https://web.archive.org/web/20140326093356/http://torrentfreak.com/publicbt-tracker-set-to-patch-bittorrents-achilles-heel-090712/) from the original on 26 March 2014. Retrieved 14 July 2009.

66. Chwan-Hwa (John) Wu, J. David Irwin. *Introduction to Computer Networks and Cybersecurity*. Chapter 5.4.: Partially Centralized Architectures. CRC Press. February 4, 2013. ISBN 9781466572133
67. Worthington, David; Nate Mook (25 May 2005). "BitTorrent Creator Opens Online Search" (http://www.betanews.com/article/BitTorrent_Creator_Opens_Online_Search/1117065427). BetaNews. Archived (https://web.archive.org/web/20090224041429/http://www.betanews.com/article/BitTorrent_Creator_Opens_Online_Search/1117065427) from the original on 24 February 2009. Retrieved 9 May 2006.
68. "Vuze Changelog" (http://azureus.sourceforge.net/changelog.php). Azureus.sourceforge.net. Archived (https://web.archive.org/web/20061201095553/http://azureus.sourceforge.net/changelog.php) from the original on 1 December 2006.
69. Wang, Liang; Kangasharju, Jussi. (2013). "Measuring Large-Scale Distributed Systems: Case of BitTorrent Mainline DHT" (http://www.cs.helsinki.fi/u/lxwang/publications/P2P2013_13.pdf) (PDF). *IEEE Peer-to-Peer*. Retrieved 15 May 2014.
70. "Khashmir.Sourceforge.net" (http://khashmir.sourceforge.net/). Khashmir.Sourceforge.net. Archived (https://web.archive.org/web/20120702140624/http://khashmir.sourceforge.net/) from the original on 2 July 2012. Retrieved 9 July 2012.
71. "Azureus.sourceforge.net" (http://azureus.sourceforge.net/plugin_details.php?plugin=mlDHT). Azureus.sourceforge.net. Archived (https://web.archive.org/web/20120801195122/http://azureus.sourceforge.net/plugin_details.php?plugin=mlDHT) from the original on 1 August 2012. Retrieved 9 July 2012.
72. "HTTP-Based Seeding Specification" (https://www.webcitation.org/6184q7Pjn?url=http://bittornado.com/docs/webseed-spec.txt). BitTornado.com. Archived from the original (http://bittornado.com/docs/webseed-spec.txt) (TXT) on 22 August 2011. Retrieved 9 May 2006.
73. John Hoffman, DeHackEd (25 February 2008). "*HTTP Seeding* – BitTorrent Enhancement Proposal № 17" (http://www.bittorrent.org/beps/bep_0017.html). Archived (https://web.archive.org/web/20131213074432/http://www.bittorrent.org/beps/bep_0017.html) from the original on 13 December 2013. Retrieved 17 February 2012.
74. "HTTP/FTP Seeding for BitTorrent" (http://www.getright.com/seedtorrent.html). GetRight.com. Archived (https://web.archive.org/web/20091228072458/http://getright.com/seedtorrent.html) from the original on 28 December 2009. Retrieved 18 March 2010.
75. Michael Burford (25 February 2008). "*WebSeed – HTTP/FTP Seeding (GetRight style)* – BitTorrent Enhancement Proposal № 19" (http://www.bittorrent.org/beps/bep_0019.html). Bittorrent.org. Archived (https://web.archive.org/web/20131213074337/http://www.bittorrent.org/beps/bep_0019.html) from the original on 13 December 2013. Retrieved 17 February 2012.
76. "Burn Any Web-Hosted File into a Torrent With Burnbit" (http://torrentfreak.com/burn-any-web-hosted-file-into-a-torrent-with-burnbit-100913/). TorrentFreak. 13 September 2010. Archived (https://web.archive.org/web/20110809053857/http://torrentfreak.com/burn-any-web-hosted-file-into-a-torrent-with-burnbit-100913/) from the original on 9 August 2011. Retrieved 9 July 2012.
77. "PHP based torrent file creator, tracker and seed server" (http://php-tracker.org/). PHPTracker. Archived (https://web.archive.org/web/20131219084439/http://php-tracker.org/) from the original on 19 December 2013. Retrieved 9 July 2012.
78. Gillmor, Steve (13 December 2003). "BitTorrent and RSS Create Disruptive Revolution" (http://www.eweek.com/c/a/Messaging-and-Collaboration/BitTorrent-and-RSS-Create-Disruptive-Revolution/). EWeek.com. Retrieved 22 April 2007.
79. Miller, Ernest (2 March 2004). "BitTorrent + RSS = The New Broadcast" (http://importance.corante.com/archives/002223.html). Archived (https://web.archive.org/web/20131023060205/http://importance.corante.com/archives/002223.html) from the original on 23 October 2013.. *The Importance of...* Corante.com.
80. Raymond, Scott (16 December 2003). "Broadcatching with BitTorrent" (https://web.archive.org/web/20040213093750/http://scottraymond.net/archive/4745). scottraymond.net. Archived from the original (http://scottraymond.net/archive/4745) on 13 February 2004.
81. "MoveDigital API REST functions" (https://web.archive.org/web/20060811154118/http://www.movedigital.com/docs/index.php/MoveDigital_API). Move Digital. 2006. Archived from the original (http://www.movedigital.com/docs/index.php/MoveDigital_API) on 11 August 2006. Retrieved 9 May 2006. Documentation.

82. "Prodigem Enclosure Puller(pep.txt)" (https://web.archive.org/web/20060526130219/http://prodigem.com/code/pep/pep.txt). Prodigem.com. Archived from the original (http://prodigem.com/code/pep/pep.txt) (TXT) on 26 May 2006. Retrieved 9 May 2006. via Internet Wayback Machine.
83. "Encrypting Bittorrent to take out traffic shapers" (http://torrentfreak.com/encrypting-bittorrent-to-take-out-traffic-shapers/). Torrentfreak.com. 5 February 2006. Archived (https://web.archive.org/web/20140326092903/http://torrentfreak.com/encrypting-bittorrent-to-take-out-traffic-shapers/) from the original on 26 March 2014. Retrieved 9 May 2006.
84. Sales, Ben (27 September 2006). "ResTech solves network issues" (http://www.studlife.com/archives/News/2006/09/27/ResTechsolvesnetworkissues/). studlife.com. Archived (https://web.archive.org/web/20140326094918/http://www.studlife.com/archives/News/2006/09/27/ResTechsolvesnetworkissues/) from the original on 26 March 2014.
85. "Comcast Throttles BitTorrent Traffic, Seeding Impossible" (http://torrentfreak.com/comcast-throttles-bittorrent-traffic-seeding-impossible/). Archived (https://web.archive.org/web/20131011044156/http://torrentfreak.com/comcast-throttles-bittorrent-traffic-seeding-impossible/) from the original on 11 October 2013., *TorrentFreak*, 17 August 2007.
86. Broache, Anne (27 March 2008). "Comcast and BitTorrent Agree to Collaborate" (http://www.news.com/8301-10784_3-9904494-7.html). News.com. Archived (https://web.archive.org/web/20080509150131/http://www.news.com/8301-10784_3-9904494-7.html) from the original on 9 May 2008. Retrieved 9 July 2012.
87. Soghoian, Chris (4 September 2007). "Is Comcast's BitTorrent filtering violating the law?" (http://www.cnet.com/8301-13739_1-9769645-46.html). Cnet.com. Archived (https://web.archive.org/web/20100715153001/http://www.cnet.com/8301-13739_1-9769645-46.html) from the original on 15 July 2010. Retrieved 9 July 2012.
88. "BEP12: Multitracker Metadata Extension" (http://www.bittorrent.org/beps/bep_0012.html). BitTorrent Inc. Archived (https://web.archive.org/web/20121227233108/http://bittorrent.org/beps/bep_0012.html) from the original on 27 December 2012. Retrieved 28 March 2013.
89. "P2P:Protocol:Specifications:Multitracker" (http://wiki.depthstrike.com/index.php/P2P:Protocol:Specifications:Multitracker#Bad_Implementations). wiki.depthstrike.com. Archived (https://web.archive.org/web/20140326095037/http://wiki.depthstrike.com/index.php/P2P%3AProtocol%3ASpecifications%3AMultitracker#Bad_Implementations) from the original on 26 March 2014. Retrieved 13 November 2009.
90. "DecentralizedRecommendation –" (https://www.tribler.org/DecentralizedRecommendation). Tribler.org. Archived (https://web.archive.org/web/20081202143338/http://www.tribler.org/DecentralizedRecommendation) from the original on 2 December 2008. Retrieved 9 July 2012.
91. Wong, Bernard; Vigfusson, Ymir; Gun Sirer, Emin (2 May 2007). "Hyperspaces for Object Clustering and Approximate Matching in Peer-to-Peer Overlays" (http://www.cs.cornell.edu/People/egs/papers/hyperspaces.pdf) (PDF). Cornell University. Retrieved 7 April 2013.
92. Wong, Bernard (2008). "Cubit: Approximate Matching for Peer-to-Peer Overlays" (http://www.cs.cornell.edu/~bwong/cubit/index.html). Cornell University. Archived (https://web.archive.org/web/20121231060445/http://www.cs.cornell.edu/~bwong/cubit/index.html) from the original on 31 December 2012. Retrieved 26 May 2008.
93. Wong, Bernard. "Approximate Matching for Peer-to-Peer Overlays with Cubit" (http://www.cs.cornell.edu/~bwong/cubit/tr-cubit.pdf) (PDF). Cornell University. Retrieved 26 May 2008.
94. "Torrent Exchange" (http://wiki.bitcomet.com/Torrent_Exchange). Archived (https://web.archive.org/web/20131005065144/http://wiki.bitcomet.com/Torrent_Exchange) from the original on 5 October 2013. Retrieved 31 January 2010. "The torrent sharing feature of BitComet. Bitcomet.com."
95. Van Der Sar, Ernesto (4 December 2009). "Thunder Blasts uTorrent's Market Share Away" (http://torrentfreak.com/thunder-blasts-utorrents-market-share-away-091204/). TorrentFreak. Archived (https://www.webcitation.org/61iuRToZn?url=http://torrentfreak.com/thunder-blasts-utorrents-market-share-away-091204/) from the original on 15 September 2011. Retrieved 15 September 2011.
96. "uTorrent Dominates BitTorrent Client Market Share" (https://torrentfreak.com/utorrent-dominates-bittorrent-client-market-share-090624/). TorrentFreak. 24 June 2009. Archived (https://web.archive.org/web/20140403051947/http://torrentfreak.com/utorrent-dominates-bittorrent-client-market-share-090624/) from the original on 3 April 2014. Retrieved 25 June 2013.

97. "Windows Public File Sharing Market Share 2015" (https://www.opswat.com/resources/reports/market-share-usage-analysis-file-sharing-antivirus-june-2015#public-file-sharing). opswat. Retrieved 1 April 2016.
98. Henry, Alan. "Most Popular BitTorrent Client 2015" (http://lifehacker.com/5813348/five-best-bittorrent-applications/1705622513). lifehacker. Retrieved 1 April 2016.
99. "Torrent Server combines a file server with P2P file sharing" (http://www.turnkeylinux.org/torrentserver). Turnkeylinux.org. Retrieved 9 July 2012.
100. Anderson, Nate (1 February 2007). "Does network neutrality mean an end to BitTorrent throttling?" (https://arstechnica.com/news.ars/post/20070201-8750.html). Ars Technica, LLC. Archived (https://web.archive.org/web/20081216140732/http://arstechnica.com/news.ars/post/20070201-8750.html) from the original on 16 December 2008. Retrieved 9 February 2007.
101. Himabindu Pucha; David G. Andersen; Michael Kaminsky (April 2007). "Exploiting Similarity for Multi-Source Downloads Using File Handprints" (http://www.cs.cmu.edu/~dga/papers/nsdi2007-set/). Purdue University, Carnegie Mellon University, Intel Research Pittsburgh. Archived (https://web.archive.org/web/20130618075650/http://www.cs.cmu.edu/~dga/papers/nsdi2007-set/) from the original on 18 June 2013. Retrieved 15 April 2007.
102. "Speed boost plan for file-sharing" (http://news.bbc.co.uk/2/hi/technology/6544919.stm). BBC News. 12 April 2007. Archived (https://web.archive.org/web/20081207163345/http://news.bbc.co.uk/2/hi/technology/6544919.stm) from the original on 7 December 2008. Retrieved 21 April 2007.
103. Johnston, Casey (9 December 2008). "Arstechnica.com" (https://arstechnica.com/news.ars/post/20081209-bittorrent-has-new-plan-to-shape-up-p2p-behavior.html). Arstechnica.com. Archived (https://web.archive.org/web/20081212061637/http://arstechnica.com/news.ars/post/20081209-bittorrent-has-new-plan-to-shape-up-p2p-behavior.html) from the original on 12 December 2008. Retrieved 9 July 2012.
104. "Is torrenting safe? Is it illegal? Are you likely to be caught?" (https://www.comparitech.com/blog/vpn-privacy/is-torrenting-safe-illegal-will-you-be-caught/). 29 November 2018.
105. "The Piratebay is Down: Raided by the Swedish Police" (http://torrentfreak.com/the-piratebay-is-down-raided-by-the-swedish-police/). TorrentFreak. 31 May 2006. Archived (https://web.archive.org/web/20140416060632/https://torrentfreak.com/the-piratebay-is-down-raided-by-the-swedish-police/) from the original on 16 April 2014. Retrieved 20 May 2007.
106. "Technical report: An Estimate of Infringing Use of the Internet" (http://documents.envisional.com/docs/Envisional-Internet_Usage-Jan2011.pdf) (PDF). Envisional. 1 January 2011. Retrieved 6 May 2012.
107. "BitTorrent: Copyright lawyers' favourite target reaches 200,000 lawsuits" (https://web.archive.org/web/20131204002125/http://www.theguardian.com/technology/pda/2011/aug/09/bittorrent-piracy). The Guardian. 9 August 2011. Archived from the original (https://www.theguardian.com/technology/pda/2011/aug/09/bittorrent-piracy) on 4 December 2013. Retrieved 10 January 2014.
108. Albanesius, Chloe (30 April 2012). "U.K. High Court Orders ISPs to Block The Pirate Bay" (https://www.pcmag.com/article2/0,2817,2403749,00.asp). PC Magazine. Archived (https://web.archive.org/web/20130525155105/http://www.pcmag.com/article2/0%2C2817%2C2403749%2C00.asp) from the original on 25 May 2013. Retrieved 6 May 2012.
109. Adamsky, Florian (2015). "P2P File-Sharing in Hell: Exploiting BitTorrent Vulnerabilities to Launch Distributed Reflective DoS Attacks" (https://www.usenix.org/conference/woot15/workshop-program/presentation/p2p-file-sharing-hell-exploiting-bittorrent).
110. Bhakuni, A; Sharma, P; Kaushal, R (2014). "Free-rider detection and punishment in Bit *Torrent* based P2P networks". *2014 IEEE International Advance Computing Conference (IACC)* (http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6779311&tag=1). International Advanced Computing Conference. p. 155. doi:10.1109/IAdCC.2014.6779311 (https://doi.org/10.1109%2FIAdCC.2014.6779311). ISBN 978-1-4799-2572-8.
111. Berns, Andrew D.; Jung, Eunjin (EJ) (24 April 2008). "Searching for Malware in Bit Torrent" (https://web.archive.org/web/20130501051349/http://www.techrepublic.com/whitepapers/searching-for-malware-in-bittorrent/1681115). University of Iowa, via TechRepublic. Archived from the original (http://www.techrepublic.com/whitepapers/searching-for-malware-in-bittorrent/1681115) on 1 May 2013. Retrieved 7 April 2013.(registration required)

112. Vegge, Håvard; Halvorsen, Finn Michael; Nergård, Rune Walsø (2009), "Where Only Fools Dare to Tread: An Empirical Study on the Prevalence of Zero-Day Malware", *2009 Fourth International Conference on Internet Monitoring and Protection* (https://web.archive.org/web/20130617143905/http://www.rookconsulting.com/Downloads/NTNU-zeroday-project-2008.pdf) (PDF), IEEE Computer Society, p. 66, doi:10.1109/ICIMP.2009.19 (https://doi.org/10.1109%2FICIMP.2009.19), ISBN 978-1-4244-3839-6, archived from the original (http://www.rookconsulting.com/Downloads/NTNU-zeroday-project-2008.pdf) (PDF *(orig. work + pub. paper)*) on 2013-06-17
113. "BitErrant" (https://biterrant.io/). *biterrant.io*. Retrieved 23 April 2017.

Further reading

- Pouwelse, Johan; et al. (2005). "The Bittorrent P2P File-Sharing System: Measurements and Analysis". *Peer-to-Peer Systems IV* (https://books.google.com/books?id=Dnw7E8xzQUMC&lpg=PA205&dq=The%20Bittorrent%20P2P%20File-Sharing%20System%3A%20Measurements%20and%20Analysis%20Johan%20Pouwelse%20%2C%20Pawel%20Garbacki%2C%20Dick%20Epema%2C&pg=PA205#v=onepage&q&f=false). Berlin: Springer. pp. 205–216. doi:10.1007/11558989_19 (https://doi.org/10.1007%2F11558989_19). ISBN 978-3-540-29068-1. Retrieved 4 September 2011.

External links

- Official website (http://www.bittorrent.org/)
- Specification (http://www.bittorrent.org/beps/bep_0003.html)
- BitTorrent (https://curlie.org/Computers/Internet/File_Sharing/BitTorrent) at Curlie
- Interview with chief executive Ashwin Navin (https://web.archive.org/web/20061230021548/http://streaming.scmp.com/podcasting/upload/News_BitTorrent_june15.mp3)
- Unofficial BitTorrent Protocol Specification v1.0 (http://wiki.theory.org/BitTorrentSpecification) at wiki.theory.org
- Unofficial BitTorrent Location-aware Protocol 1.0 Specification (http://wiki.theory.org/BitTorrent_Location-aware_Protocol_1.0_Specification) at wiki.theory.org
- Czerniawski, Michal (20 December 2009). "Responsibility of Bittorrent Search Engines for Copyright Infringements". SSRN. doi:10.2139/ssrn.1540913 (https://doi.org/10.2139%2Fssrn.1540913). SSRN 1540913 (https://ssrn.com/abstract=1540913). Missing or empty `|url=` (help)
- Cohen, Bram (16 February 2005). "Under the hood of BitTorrent" (http://www.stanford.edu/class/ee380/Abstracts/050216.html). *Computer Systems Colloquium (EE380)*. Stanford University.

WIKIPEDIA

Voice over IP

Voice over Internet Protocol (**VoIP**), also called **IP telephony**, is a methodology and group of technologies for the delivery of voice communications and multimedia sessions over Internet Protocol (IP) networks, such as the Internet. The terms **Internet telephony**, **broadband telephony**, and **broadband phone service** specifically refer to the provisioning of communications services (voice, fax, SMS, voice-messaging) over the public Internet, rather than via the public switched telephone network (PSTN).

The steps and principles involved in originating VoIP telephone calls are similar to traditional digital telephony and involve signaling, channel setup, digitization of the analog voice signals, and encoding. Instead of being transmitted over a circuit-switched network, the digital information is packetized, and transmission occurs as IP packets over a packet-switched network. They transport media streams using special media delivery protocols that encode audio and video with audio codecs, and video codecs. Various codecs exist that optimize the media stream based on application requirements and network bandwidth; some implementations rely on narrowband and compressed speech, while others support high-fidelity stereo codecs. Some popular codecs include μ-law and A-law versions of G.711, G.722, an open source voice codec known as iLBC, a codec that uses only 8 kbit/s each way called G.729, and many others.

Early providers of voice-over-IP services offered business models and technical solutions that mirrored the architecture of the legacy telephone network. Second-generation providers, such as Skype, built closed networks for private user bases, offering the benefit of free calls and convenience while potentially charging for access to other communication networks, such as the PSTN. This limited the freedom of users to mix-and-match third-party hardware and software. Third-generation providers, such as Google Talk, adopted the concept of federated VoIP—which is a departure from the architecture of the legacy networks.[1] These solutions typically allow dynamic interconnection between users on any two domains on the Internet when a user wishes to place a call.

In addition to VoIP phones, VoIP is also available on many personal computers and other Internet access devices. Calls and SMS text messages may be sent over mobile data or Wi-Fi.[2] VoIP allows modern communications technologies (including telephones, smartphones, voice and video conferencing, email, and presence detection) to be consolidated using a single unified communications system.[3]

Contents

Pronunciation

VoIP is variously pronounced as an initialism, *V-O-I-P*, or as an acronym /ˈvɔɪp/ (*voyp*), as in *voice*.[4] Full words, *voice over Internet Protocol*, or *voice over IP*, are sometimes used.

Protocols

Voice over IP has been implemented in various ways using both proprietary protocols and protocols based on open standards. These protocols can be used by a VoIP phone, special-purpose software, a mobile application or integrated into a web page. VoIP protocols include:

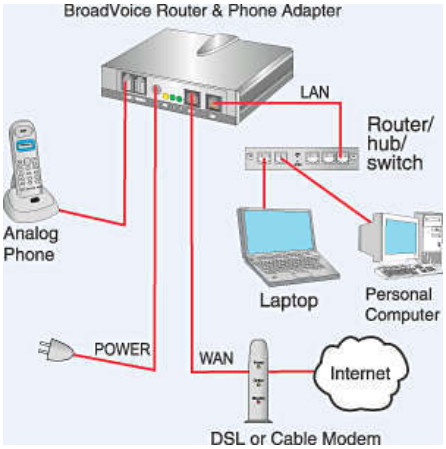
- Session Initiation Protocol (SIP), connection management protocol developed by the IETF
- H.323, one of the first VoIP call signaling and control protocols that found widespread implementation. Since the development of newer, less complex protocols such as MGCP and SIP, H.323 deployments are increasingly limited to carrying existing long-haul network traffic.
- Media Gateway Control Protocol (MGCP), connection management for media gateways
- H.248, control protocol for media gateways across a converged internetwork consisting of the traditional public switched telephone network (PSTN) and modern packet networks
- Real-time Transport Protocol (RTP), transport protocol for real-time audio and video data
- Real-time Transport Control Protocol (RTCP), sister protocol for RTP providing stream statistics and status information
- Secure Real-time Transport Protocol (SRTP), encrypted version of RTP
- Session Description Protocol (SDP), file format used principally by SIP to describe VoIP connections
- Inter-Asterisk eXchange (IAX), protocol used between VoIP servers

- Extensible Messaging and Presence Protocol (XMPP), instant messaging, presence information, and contact list maintenance
- Jingle, adds peer-to-peer session control to XMPP
- Skype protocol, proprietary Internet telephony protocol suite based on peer-to-peer architecture

Adoption

Consumer market

Mass-market VoIP services use existing broadband Internet access, by which subscribers place and receive telephone calls in much the same manner as they would via the public switched telephone network (PSTN). Full-service VoIP phone companies provide inbound and outbound service with direct inbound dialing. Many offer unlimited domestic calling and sometimes international calls for a flat monthly subscription fee. Phone calls between subscribers of the same provider are usually free when flat-fee service is not available.



Example of residential network including VoIP

A VoIP phone is necessary to connect to a VoIP service provider. This can be implemented in several ways:

- Dedicated VoIP phones connect directly to the IP network using technologies such as wired Ethernet or Wi-Fi. These are typically designed in the style of traditional digital business telephones.
- An analog telephone adapter connects to the network and implements the electronics and firmware to operate a conventional analog telephone attached through a modular phone jack. Some residential Internet gateways and cablemodems have this function built in.
- Softphone application software installed on a networked computer that is equipped with a microphone and speaker, or headset. The application typically presents a dial pad and display field to the user to operate the application by mouse clicks or keyboard input.

PSTN and mobile network providers

It is increasingly common for telecommunications providers to use VoIP telephony over dedicated and public IP networks as a backhaul to connect switching centers and to interconnect with other telephony network providers; this is often referred to as *IP backhaul*.[5][6]

Smartphones may have SIP clients built into the firmware or available as an application download.[7][8]

Corporate use

Because of the bandwidth efficiency and low costs that VoIP technology can provide, businesses are migrating from traditional copper-wire telephone systems to VoIP systems to reduce their monthly phone costs. In 2008, 80% of all new Private branch exchange (PBX) lines installed internationally were VoIP.[9] For example, in the United States, the Social Security Administration is converting its field offices of 63,000 workers from traditional phone installations to a VoIP infrastructure carried over its existing data network.[10][11]

VoIP allows both voice and data communications to be run over a single network, which can significantly reduce infrastructure costs. The prices of extensions on VoIP are lower than for PBX and key systems. VoIP switches may run on commodity hardware, such as personal computers. Rather than closed architectures, these devices rely on standard interfaces.[12] VoIP devices have simple, intuitive user interfaces, so users can often make simple system configuration

changes. Dual-mode phones enable users to continue their conversations as they move between an outside cellular service and an internal Wi-Fi network, so that it is no longer necessary to carry both a desktop phone and a cell phone. Maintenance becomes simpler as there are fewer devices to oversee.[12]

VoIP solutions aimed at businesses have evolved into unified communications services that treat all communications—phone calls, faxes, voice mail, e-mail, web conferences, and more—as discrete units that can all be delivered via any means and to any handset, including cellphones. Two kinds of service providers are operating in this space: one set is focused on VoIP for medium to large enterprises, while another is targeting the small-to-medium business (SMB) market.[13]

Skype, which originally marketed itself as a service among friends, has begun to cater to businesses, providing free-of-charge connections between any users on the Skype network and connecting to and from ordinary PSTN telephones for a charge.[14]

Quality of service

Communication on the IP network is perceived as less reliable in contrast to the circuit-switched public telephone network because it does not provide a network-based mechanism to ensure that data packets are not lost, and are delivered in sequential order. It is a best-effort network without fundamental Quality of Service (QoS) guarantees. Voice, and all other data, travels in packets over IP networks with fixed maximum capacity. This system may be more prone to data loss in the presence of congestion[a] than traditional circuit switched systems; a circuit switched system of insufficient capacity will refuse new connections while carrying the remainder without impairment, while the quality of real-time data such as telephone conversations on packet-switched networks degrades dramatically.[16] Therefore, VoIP implementations may face problems with latency, packet loss, and jitter.[16][17]

By default, network routers handle traffic on a first-come, first-served basis. Fixed delays cannot be controlled as they are caused by the physical distance the packets travel. They are especially problematic when satellite circuits are involved because of the long distance to a geostationary satellite and back; delays of 400–600 ms are typical. Latency can be minimized by marking voice packets as being delay-sensitive with QoS methods such as DiffServ.[16]

Network routers on high volume traffic links may introduce latency that exceeds permissible thresholds for VoIP. Excessive load on a link can cause congestion and associated queueing delays and packet loss. This signals a transport protocol like TCP to reduce its transmission rate to alleviate the congestion. But VoIP usually uses UDP not TCP because recovering from congestion through retransmission usually entails too much latency.[16] So QoS mechanisms can avoid the undesirable loss of VoIP packets by immediately transmitting them ahead of any queued bulk traffic on the same link, even when the link is congested by bulk traffic.

VoIP endpoints usually have to wait for completion of transmission of previous packets before new data may be sent. Although it is possible to preempt (abort) a less important packet in mid-transmission, this is not commonly done, especially on high-speed links where transmission times are short even for maximum-sized packets.[18] An alternative to preemption on slower links, such as dialup and digital subscriber line (DSL), is to reduce the maximum transmission time by reducing the maximum transmission unit. But since every packet must contain protocol headers, this increases relative header overhead on every link traversed.[18]

The receiver must resequence IP packets that arrive out of order and recover gracefully when packets arrive too late or not at all. Packet delay variation results from changes in queuing delay along a given network path due to competition from other users for the same transmission links. VoIP receivers accommodate this variation by storing incoming packets briefly in a playout buffer, deliberately increasing latency to improve the chance that each packet will be on hand when it is time for the voice engine to play it. The added delay is thus a compromise between excessive latency and excessive dropout, i.e. momentary audio interruptions.

Although jitter is a random variable, it is the sum of several other random variables which are at least somewhat independent: the individual queuing delays of the routers along the Internet path in question. Motivated by the central limit theorem, jitter can be modeled as a gaussian random variable. This suggests continually estimating the mean delay and its standard deviation and setting the playout delay so that only packets delayed more than several standard deviations above the mean will arrive too late to be useful. In practice, the variance in latency of many Internet paths is dominated by a small number (often one) of relatively slow and congested bottleneck links. Most Internet backbone links are now so fast (e.g. 10 Gbit/s) that their delays are dominated by the transmission medium (e.g. optical fiber) and the routers driving them do not have enough buffering for queuing delays to be significant. Some VoIP providers now offer fiber to the building (FTTB) in order to ensure the highest VoIP call quality, reliability, and service up-time.[3]

It has been suggested to rely on the packetized nature of media in VoIP communications and transmit the stream of packets from the source phone to the destination phone simultaneously across different routes (multi-path routing).[19] In such a way, temporary failures have less impact on the communication quality. In capillary routing at the packet level Fountain codes or particularly raptor codes it is recommended for transmitting extra redundant packets making the communication more reliable.

A number of protocols have been defined to support the reporting of quality of service (QoS) and quality of experience (QoE) for VoIP calls. These include RTCP Extended Report (RFC 3611[20]), SIP RTCP Summary Reports, H.460.9 Annex B (for H.323), H.248.30 and MGCP extensions. The RFC 3611 VoIP Metrics block is generated by an IP phone or gateway during a live call and contains information on packet loss rate, packet discard rate (because of jitter), packet loss/discard burst metrics (burst length/density, gap length/density), network delay, end system delay, signal/noise/echo level, Mean Opinion Scores (MOS) and R factors and configuration information related to the jitter buffer.

RFC 3611[20] VoIP metrics reports are exchanged between IP endpoints on an occasional basis during a call, and an end of call message sent via SIP RTCP Summary Report or one of the other signaling protocol extensions. RFC 3611 VoIP metrics reports are intended to support real time feedback related to QoS problems, the exchange of information between the endpoints for improved call quality calculation and a variety of other applications.

Rural areas in particular are greatly hindered in their ability to choose a VoIP system over PBX. This is generally down to the poor access to superfast broadband in rural country areas. With the release of 4G data, there is a potential for corporate users based outside of populated areas to switch their internet connection to 4G data, which is comparatively as fast as a regular superfast broadband connection. This greatly enhances the overall quality and user experience of a VoIP system in these areas.

DSL and ATM

DSL modems provide Ethernet (or Ethernet over USB) connections to local equipment, but inside they are actually Asynchronous Transfer Mode (ATM) modems (Note: Non-ATM technologies such as 802.3ah also provide this capability). They use ATM Adaptation Layer 5 (AAL5) to segment each Ethernet packet into a series of 53-byte ATM cells for transmission, reassembling them back into Ethernet frames at the receiving end. A virtual circuit identifier (VCI) is part of the 5-byte header on every ATM cell, so the transmitter can multiplex the active virtual circuits (VCs) in any arbitrary order. Cells from the *same* VC are always sent sequentially.

A majority of DSL providers use only one VC for each customer, even those with bundled VoIP service. Every Ethernet frame must be completely transmitted before another can begin. If a second VC were established, given high priority and reserved for VoIP, then a low priority data packet could be suspended in mid-transmission and a VoIP packet sent right away on the high priority VC. Then the link would pick up the low priority VC where it left off. Because ATM links

are multiplexed on a cell-by-cell basis, a high priority packet would have to wait at most 53 byte times to begin transmission. There would be no need to reduce the interface MTU and accept the resulting increase in higher layer protocol overhead, and no need to abort a low priority packet and resend it later.

ATM has substantial header overhead: 5/53 = 9.4%, roughly twice the total header overhead of a 1500 byte Ethernet frame. This "ATM tax" is incurred by every DSL user whether or not they take advantage of multiple virtual circuits - and few can.[16]

ATM's potential for latency reduction is greatest on slow links, because worst-case latency decreases with increasing link speed. A full-size (1500 byte) Ethernet frame takes 94 ms to transmit at 128 kbit/s but only 8 ms at 1.5 Mbit/s. If this is the bottleneck link, this latency is probably small enough to ensure good VoIP performance without MTU reductions or multiple ATM VCs. The latest generations of DSL, VDSL and VDSL2, carry Ethernet without intermediate ATM/AAL5 layers, and they generally support IEEE 802.1p priority tagging so that VoIP can be queued ahead of less time-critical traffic.[16]

Layer 2

A number of protocols that deal with the data link layer and physical layer include quality-of-service mechanisms that can be used to ensure that applications like VoIP work well even in congested scenarios. Some examples include:

- IEEE 802.11e is an approved amendment to the IEEE 802.11 standard that defines a set of quality-of-service enhancements for wireless LAN applications through modifications to the Media Access Control (MAC) layer. The standard is considered of critical importance for delay-sensitive applications, such as voice over wireless IP.
- IEEE 802.1p defines 8 different classes of service (including one dedicated to voice) for traffic on layer-2 wired Ethernet.
- The ITU-T G.hn standard, which provides a way to create a high-speed (up to 1 gigabit per second) Local area network (LAN) using existing home wiring (power lines, phone lines and coaxial cables). G.hn provides QoS by means of "Contention-Free Transmission Opportunities" (CFTXOPs) which are allocated to flows (such as a VoIP call) which require QoS and which have negotiated a "contract" with the network controllers.

Performance metrics

The quality of voice transmission is characterized by several metrics that may be monitored by network elements, by the user agent hardware or software. Such metrics include network packet loss, packet jitter, packet latency (delay), post-dial delay, and echo. The metrics are determined by VoIP performance testing and monitoring.[21][22][23][24][25][26]

PSTN integration

A VoIP media gateway controller (aka Class 5 Softswitch) works in cooperation with a media gateway (aka IP Business Gateway) and connects the digital media stream, so as to complete creating the path for voice as well as data media. They include the interfaces for connecting the standard PSTN networks with the ATM and Inter Protocol networks. The Ethernet interfaces are also included in the modern systems, which are specially designed to link calls that are passed via the VoIP.[27]

E.164 is a global FGFnumbering standard for both the PSTN and PLMN. Most VoIP implementations support E.164 to allow calls to be routed to and from VoIP subscribers and the PSTN/PLMN.[28] VoIP implementations can also allow other identification techniques to be used. For example, Skype allows subscribers to choose "Skype names"[29] (usernames) whereas SIP implementations can use URIs[30] similar to email addresses. Often VoIP implementations employ methods of translating non-E.164 identifiers to E.164 numbers and vice versa, such as the Skype-In service provided by Skype[31] and the ENUM service in IMS and SIP.[32]

Echo can also be an issue for PSTN integration.[33] Common causes of echo include impedance mismatches in analog circuitry and acoustic coupling of the transmit and receive signal at the receiving end.

Number portability

Local number portability (LNP) and mobile number portability (MNP) also impact VoIP business. In November 2007, the Federal Communications Commission in the United States released an order extending number portability obligations to interconnected VoIP providers and carriers that support VoIP providers.[34] Number portability is a service that allows a subscriber to select a new telephone carrier without requiring a new number to be issued. Typically, it is the responsibility of the former carrier to "map" the old number to the undisclosed number assigned by the new carrier. This is achieved by maintaining a database of numbers. A dialed number is initially received by the original carrier and quickly rerouted to the new carrier. Multiple porting references must be maintained even if the subscriber returns to the original carrier. The FCC mandates carrier compliance with these consumer-protection stipulations.

A voice call originating in the VoIP environment also faces challenges to reach its destination if the number is routed to a mobile phone number on a traditional mobile carrier. VoIP has been identified in the past as a Least Cost Routing (LCR) system, which is based on checking the destination of each telephone call as it is made, and then sending the call via the network that will cost the customer the least. This rating is subject to some debate given the complexity of call routing created by number portability. With GSM number portability now in place, LCR providers can no longer rely on using the network root prefix to determine how to route a call. Instead, they must now determine the actual network of every number before routing the call.

Therefore, VoIP solutions also need to handle MNP when routing a voice call. In countries without a central database, like the UK, it might be necessary to query the GSM network about which home network a mobile phone number belongs to. As the popularity of VoIP increases in the enterprise markets because of least cost routing options, it needs to provide a certain level of reliability when handling calls.

MNP checks are important to assure that this quality of service is met. Handling MNP lookups before routing a call provides some assurance that the voice call will actually work.

Emergency calls

A telephone connected to a land line has a direct relationship between a telephone number and a physical location, which is maintained by the telephone company and available to emergency responders via the national emergency response service centers in form of emergency subscriber lists. When an emergency call is received by a center the location is automatically determined from its databases and displayed on the operator console.

In IP telephony, no such direct link between location and communications end point exists. Even a provider having hardware infrastructure, such as a DSL provider, may know only the approximate location of the device, based on the IP address allocated to the network router and the known service address. Some ISPs do not track the automatic assignment of IP addresses to customer equipment.[35]

IP communication provides for device mobility. For example, a residential broadband connection may be used as a link to a virtual private network of a corporate entity, in which case the IP address being used for customer communications may belong to the enterprise, not being the IP address of the residential ISP. Such off-premises extensions may appear as part of an upstream IP PBX. On mobile devices, e.g., a 3G handset or USB wireless broadband adapter, the IP address has no relationship with any physical location known to the telephony service provider, since a mobile user could be anywhere in a region with network coverage, even roaming via another cellular company.

At the VoIP level, a phone or gateway may identify itself with a Session Initiation Protocol (SIP) registrar by its account credentials. In such cases, the Internet telephony service provider (ITSP) knows only that a particular user's equipment is active. Service providers often provide emergency response services by agreement with the user who registers a physical location and agrees that emergency services are provided to that address only if an emergency number is called from the IP device.

Such emergency services are provided by VoIP vendors in the United States by a system called Enhanced 911 (E911), based on the Wireless Communications and Public Safety Act of 1999. The VoIP E911 emergency-calling system associates a physical address with the calling party's telephone number. All VoIP providers that provide access to the public switched telephone network are required to implement E911,[35] a service for which the subscriber may be charged. "VoIP providers may not allow customers to "opt-out" of 911 service."[35]

The VoIP E911 system is based on a static table lookup. Unlike in cellular phones, where the location of an E911 call can be traced using assisted GPS or other methods, the VoIP E911 information is accurate only if subscribers, who have the legal responsibility, keep their emergency address information current.

Fax support

Sending faxes over VoIP networks is sometimes referred to as Fax over IP (FoIP). Transmission of fax documents was problematic in early VoIP implementations, as most voice digitization and compression codecs are optimized for the representation of the human voice and the proper timing of the modem signals cannot be guaranteed in a packet-based, connection-less network. A standards-based solution for reliably delivering fax-over-IP is the T.38 protocol.

The T.38 protocol is designed to compensate for the differences between traditional packet-less communications over analog lines and packet-based transmissions which are the basis for IP communications. The fax machine may be a standard device connected to an analog telephone adapter (ATA), or it may be a software application or dedicated network device operating via an Ethernet interface.[36] Originally, T.38 was designed to use UDP or TCP transmission methods across an IP network. UDP provides near real-time characteristics due to the "no recovery rule" when a UDP packet is lost or an error occurs during transmission.[37]

Some newer high end fax machines have built-in T.38 capabilities which are connected directly to a network switch or router. In T.38 each packet contains a portion of the data stream sent in the previous packet. Two successive packets have to be lost to actually lose data integrity.

Power requirements

Telephones for traditional residential analog service are usually connected directly to telephone company phone lines which provide direct current to power most basic analog handsets independently of locally available electrical power.

IP Phones and VoIP telephone adapters connect to routers or cable modems which typically depend on the availability of mains electricity or locally generated power.[38] Some VoIP service providers use customer premises equipment (e.g., cablemodems) with battery-backed power supplies to assure uninterrupted service for up to several hours in case of local power failures. Such battery-backed devices typically are designed for use with analog handsets.

Some VoIP service providers implement services to route calls to other telephone services of the subscriber, such a cellular phone, in the event that the customer's network device is inaccessible to terminate the call.

The susceptibility of phone service to power failures is a common problem even with traditional analog service in areas where many customers purchase modern telephone units that operate with wireless handsets to a base station, or that have other modern phone features, such as built-in voicemail or phone book features.

Security

The security concerns of VoIP telephone systems are similar to those of other Internet-connected devices. This means that hackers with knowledge of VoIP vulnerabilities can perform denial-of-service attacks, harvest customer data, record conversations, and compromise voicemail messages. Compromised VoIP user account or session credentials may enable an attacker to incur substantial charges from third-party services, such as long-distance or international calling.

The technical details of many VoIP protocols create challenges in routing VoIP traffic through firewalls and network address translators, used to interconnect to transit networks or the Internet. Private session border controllers are often employed to enable VoIP calls to and from protected networks. Other methods to traverse NAT devices involve assistive protocols such as STUN and Interactive Connectivity Establishment (ICE).

Though many consumer VoIP solutions do not support encryption of the signaling path or the media, securing a VoIP phone is conceptually easier to implement than on traditional telephone circuits. A result of the lack of encryption is that it is relatively easy to eavesdrop on VoIP calls when access to the data network is possible.[39] Free open-source solutions, such as Wireshark, facilitate capturing VoIP conversations.

Standards for securing VoIP are available in the Secure Real-time Transport Protocol (SRTP) and the ZRTP protocol for analog telephony adapters, as well as for some softphones. IPsec is available to secure point-to-point VoIP at the transport level by using opportunistic encryption.

Government and military organizations use various security measures to protect VoIP traffic, such as voice over secure IP (VoSIP), secure voice over IP (SVoIP), and secure voice over secure IP (SVoSIP).[40] The distinction lies in whether encryption is applied in the telephone endpoint or in the network.[41] Secure voice over secure IP may be implemented by encrypting the media with protocols such as SRTP and ZRTP. Secure voice over IP uses Type 1 encryption on a classified network, such as SIPRNet.[42][43][44][45] Public Secure VoIP is also available with free GNU software and in many popular commercial VoIP programs via libraries, such as ZRTP.[46]

Caller ID

Voice over IP protocols and equipment provide caller ID support that is compatible with the facility provided in the public switched telephone network (PSTN). Many VoIP service providers also allow callers to configure arbitrary caller ID information.[47]

Compatibility with traditional analog telephone sets

Most analog telephone adapters do not decode dial pulses generated by rotary dial telephones, but rather support only touch-tone signaling, but pulse-to-tone converters are commercially available.

Support for other telephony devices

Some special telephony services, such as those that operate in conjunction with digital video recorders, satellite television receivers, alarm systems, conventional modems over PSTN lines, may be impaired when operated over VoIP services, because of incompatibilities in design.

Operational cost

VoIP has drastically reduced the cost of communication by sharing network infrastructure between data and voice.[48][49] A single broad-band connection has the ability to transmit more than one telephone call. Secure calls using standardized protocols, such as Secure Real-time Transport Protocol, as most of the facilities of creating a secure

telephone connection over traditional phone lines, such as digitizing and digital transmission, are already in place with VoIP. It is necessary only to encrypt and authenticate the existing data stream. Automated software, such as a virtual PBX, may eliminate the need of personnel to greet and switch incoming calls.

Regulatory and legal issues

As the popularity of VoIP grows, governments are becoming more interested in regulating VoIP in a manner similar to PSTN services.[50]

Throughout the developing world, particularly in countries where regulation is weak or captured by the dominant operator, restrictions on the use of VoIP are often imposed, including in Panama where VoIP is taxed, Guyana where VoIP is prohibited and India where its retail commercial sales is allowed but only for long distance service.[51] In Ethiopia, where the government is nationalising telecommunication service, it is a criminal offence to offer services using VoIP. The country has installed firewalls to prevent international calls being made using VoIP. These measures were taken after the popularity of VoIP reduced the income generated by the state owned telecommunication company.

Canada

In Canada, the Canadian Radio-television and Telecommunications Commission regulates telephone service, including VoIP telephony service. VoIP services operating in Canada are required to provide 9-1-1 emergency service.[52]

European Union

In the European Union, the treatment of VoIP service providers is a decision for each national telecommunications regulator, which must use competition law to define relevant national markets and then determine whether any service provider on those national markets has "significant market power" (and so should be subject to certain obligations). A general distinction is usually made between VoIP services that function over managed networks (via broadband connections) and VoIP services that function over unmanaged networks (essentially, the Internet).

The relevant EU Directive is not clearly drafted concerning obligations which can exist independently of market power (e.g., the obligation to offer access to emergency calls), and it is impossible to say definitively whether VoIP service providers of either type are bound by them. A review of the EU Directive is under way and should be complete by 2007.

Arab states of the GCC

Oman

In Oman, it is illegal to provide or use unauthorized VoIP services, to the extent that web sites of unlicensed VoIP providers have been blocked. Violations may be punished with fines of 50,000 Omani Rial (about 130,317 US dollars) or spend two years in jail or both. In 2009, police raided 121 Internet cafes throughout the country and arrested 212 people for using or providing VoIP services.[53]

Saudi Arabia

In September 2017, Saudi Arabia lifted the ban on VoIPs, in an attempt to reduce operational costs and spur digital entrepreneurship.[54][55]

United Arab Emirates

In the United Arab Emirates (UAE), it is illegal to provide or use unauthorized VoIP services, to the extent that web sites of unlicensed VoIP providers have been blocked. However, some VoIPs such as Skype were allowed.[56] In January 2018, internet service providers in UAE blocked all VoIP apps, including Skype, but permitting only 2 "government-approved" VoIP apps (C'ME and BOTIM) for a fixed rate of Dh52.50 a month for use on mobile devices, and Dh105 a month to use over a computer connected."[57][58] In opposition, a petition on *Change.org* garnered over 5000 signatures, in response to which the website was blocked in UAE.[59]

India

In India, it is legal to use VoIP, but it is illegal to have VoIP gateways inside India.[60] This effectively means that people who have PCs can use them to make a VoIP call to any number, but if the remote side is a normal phone, the gateway that converts the VoIP call to a POTS call is not permitted by law to be inside India. Foreign based VoIP server services are illegal to use in India.[60]

In the interest of the Access Service Providers and International Long Distance Operators the Internet telephony was permitted to the ISP with restrictions. Internet Telephony is considered to be different service in its scope, nature and kind from real time voice as offered by other Access Service Providers and Long Distance Carriers. Hence the following type of Internet Telephony are permitted in India:[61]

> (a) PC to PC; within or outside India
> (b) PC / a device / Adapter conforming to standard of any international agencies like-ITU or IETF etc. in India to PSTN/PLMN abroad.
> (c) Any device / Adapter conforming to standards of International agencies like ITU, IETF etc. connected to ISP node with static IP address to similar device / Adapter; within or outside India.
> (d) Except whatever is described in condition (ii) above, no other form of Internet Telephony is permitted.
> (e) In India no Separate Numbering Scheme is provided to the Internet Telephony. Presently the 10 digit Numbering allocation based on E.164 is permitted to the Fixed Telephony, GSM, CDMA wireless service. For Internet Telephony the numbering scheme shall only conform to IP addressing Scheme of Internet Assigned Numbers Authority (IANA). Translation of E.164 number / private number to IP address allotted to any device and vice versa, by ISP to show compliance with IANA numbering scheme is not permitted.
> (f) The Internet Service Licensee is not permitted to have PSTN/PLMN connectivity. Voice communication to and from a telephone connected to PSTN/PLMN and following E.164 numbering is prohibited in India.

South Korea

In South Korea, only providers registered with the government are authorized to offer VoIP services. Unlike many VoIP providers, most of whom offer flat rates, Korean VoIP services are generally metered and charged at rates similar to terrestrial calling. Foreign VoIP providers encounter high barriers to government registration. This issue came to a head in 2006 when Internet service providers providing personal Internet services by contract to United States Forces Korea members residing on USFK bases threatened to block off access to VoIP services used by USFK members as an economical way to keep in contact with their families in the United States, on the grounds that the service members' VoIP providers were not registered. A compromise was reached between USFK and Korean telecommunications officials in January 2007, wherein USFK service members arriving in Korea before June 1, 2007, and subscribing to

the ISP services provided on base may continue to use their US-based VoIP subscription, but later arrivals must use a Korean-based VoIP provider, which by contract will offer pricing similar to the flat rates offered by US VoIP providers.[62]

United States

In the United States, the Federal Communications Commission requires all interconnected VoIP service providers to comply with requirements comparable to those for traditional telecommunications service providers.[63] VoIP operators in the US are required to support local number portability; make service accessible to people with disabilities; pay regulatory fees, universal service contributions, and other mandated payments; and enable law enforcement authorities to conduct surveillance pursuant to the Communications Assistance for Law Enforcement Act (CALEA).

Operators of "Interconnected" VoIP (fully connected to the PSTN) are mandated to provide Enhanced 911 service without special request, provide for customer location updates, clearly disclose any limitations on their E-911 functionality to their consumers, obtain affirmative acknowledgements of these disclosures from all consumers,[64] and 'may not allow their customers to “opt-out” of 911 service.'[65] VoIP operators also receive the benefit of certain US telecommunications regulations, including an entitlement to interconnection and exchange of traffic with incumbent local exchange carriers via wholesale carriers. Providers of "nomadic" VoIP service—those who are unable to determine the location of their users—are exempt from state telecommunications regulation.[66]

Another legal issue that the US Congress is debating concerns changes to the Foreign Intelligence Surveillance Act. The issue in question is calls between Americans and foreigners. The National Security Agency (NSA) is not authorized to tap Americans' conversations without a warrant—but the Internet, and specifically VoIP does not draw as clear a line to the location of a caller or a call's recipient as the traditional phone system does. As VoIP's low cost and flexibility convinces more and more organizations to adopt the technology, the surveillance for law enforcement agencies becomes more difficult. VoIP technology has also increased security concerns because VoIP and similar technologies have made it more difficult for the government to determine where a target is physically located when communications are being intercepted, and that creates a whole set of new legal challenges.[67]

History

The early developments of packet network designs by Paul Baran and other researchers were motivated by a desire for a higher degree of circuit redundancy and network availability in face of infrastructure failures than was possible in the circuit-switched networks in telecommunications in the mid-twentieth century. In 1973, Danny Cohen first demonstrated a form of packet voice as part of a flight simulator application, which operated across the early ARPANET.[68][69] In the following time span of about two decades, various forms of packet telephony were developed and industry interest groups formed to support the new technologies. Following the termination of the ARPANET project, and expansion of the Internet for commercial traffic, IP telephony became an established area of interest in commercial labs of the major IT concerns, such Microsoft and Intel, and open-source software, such as VocalTec, became available by the mid-1990s. By the late 1990s, the first softswitches became available, and new protocols, such as H.323, the Media Gateway Control Protocol (MGCP) and the Session Initiation Protocol (SIP) gained widespread attention. In the early 2000s, the proliferation of high-bandwidth always-on Internet connections to residential dwellings and businesses, spawned an industry of Internet telephony service providers (ITSPs). The development of open-source telephony software, such as Asterisk PBX, fueled widespread interest and entrepreneurship in voice-over-IP services, applying new Internet technology paradigms, such as cloud services to telephony.

Milestones

- 1973: Packet voice application by Danny Cohen

- 1974: The Institute of Electrical and Electronic Engineers (IEEE) publishes a paper entitled "A Protocol for Packet Network Interconnection".[70]
- 1974: Network Voice Protocol (NVP) tested over ARPANET in August 1974, carrying 16k CVSD encoded voice.
- 1977: Danny Cohen and Jon Postel of the USC Information Sciences Institute, and Vint Cerf of the Defense Advanced Research Projects Agency (DARPA), agree to separate IP from TCP, and create UDP for carrying real-time traffic.
- 1981: IPv4 is described in RFC 791.
- 1985: The National Science Foundation commissions the creation of NSFNET.[71]
- 1986: Proposals from various standards organizations for Voice over ATM, in addition to commercial packet voice products from companies such as StrataCom
- 1991: First Voice-over-IP application, Speak Freely, is released into the public domain. It was originally written by John Walker and further developed by Brian C. Wiles.[72]
- 1992: The Frame Relay Forum conducts development of standards for Voice over Frame Relay.
- 1992: InSoft Inc. announces and launches its desktop conferencing product Communique, which included VoIP and video.[73] The company is credited with developing the first generation of commercial, US-based VoIP, Internet media streaming and real-time Internet telephony/collaborative software and standards that would provide the basis for the Real Time Streaming Protocol (RTSP) standard.[74][75]
- 1994: MTALK, a freeware VoIP application for Linux[76]
- 1995: VocalTec releases *Internet Phone* commercial Internet phone software.[77][78]
  - Beginning in 1995, Intel, Microsoft and Radvision initiated standardization activities for VoIP communications system.[79]
- 1996:
  - ITU-T begins development of standards for the transmission and signaling of voice communications over Internet Protocol networks with the H.323 standard.[80]
  - US telecommunication companies petition the US Congress to ban Internet phone technology.[81]
- 1997: Level 3 began development of its first softswitch, a term they coined in 1998.[82]
- 1999:
  - The Session Initiation Protocol (SIP) specification RFC 2543 is released.[83]
  - Mark Spencer of Digium develops the first open source private branch exchange (PBX) software (Asterisk).[84]
- 2004: Commercial VoIP service providers proliferate.
- 2007: VoIP device manufacturers and sellers boom in Asia, specifically in the Philippines where many families of overseas workers reside.[85]
- 2011: Rise of WebRTC technology which allows VoIP directly in browsers

See also

- Audio over IP
- Communications Assistance For Law Enforcement Act
- Comparison of audio network protocols
- Comparison of VoIP software
- Differentiated services
- High bit rate audio video over Internet Protocol
- Integrated services
- Internet fax
- IP Multimedia Subsystem
- List of VoIP companies
- Mobile VoIP
- Network Voice Protocol
- RTP audio video profile
- SIP Trunking
- UNIStim
- Voice VPN
- VoiceXML
- VoIP recording

Notes

a. IP networks may also be more prone to DoS attacks that cause congestion.[15]

References

1. "XMPP Federation" (http://googletalk.blogspot.com/2006/01/xmpp-federation.html). Google Talkabout. 2006. Retrieved 2012-05-11.
2. Booth, C (2010). "Chapter 2: IP Phones, Software VoIP, and Integrated and Mobile VoIP". *Library Technology Reports*. **46** (5): 11–19.
3. Nespeca, Claudio (August 18, 2018). "What Is VoIP" (https://www.epiknetworks.com/what-is-voip/). *Epik Networks*. Retrieved August 18, 2018.
4. "VoIP" (http://dictionary.cambridge.org/pronunciation/british/voip). *Cambridge Dictionaries Online*.
5. "WIRELESS: Carriers look to IP for backhaul" (https://web.archive.org/web/20110809025050/http://www.eetimes.com/electronics-news/4052152/WIRELESS-Carriers-look-to-IP-for-back-haul). *www.eetimes.com*. EE Times. Archived from the original on August 9, 2011. Retrieved 8 April 2015.
6. "Mobile's IP challenge" (https://web.archive.org/web/20060217064211/http://www.totaltele.com/View.aspx?ID=77588&t=4). *www.totaltele.com*. Total Telecom Online. Archived from the original on February 17, 2006. Retrieved 8 April 2015.
7. "Android SIP Client" (https://www.callcentric.com/support/device/android/sip_client). Retrieved 2018-01-30.
8. "Learn to Make Free or Inexpensive Calls Using SIP on Android" (https://www.lifewire.com/apps-for-sip-on-android-3426471). Retrieved 2018-01-30.
9. Michael Dosch and Steve Church. "VoIP in the Broadcast Studio" (https://web.archive.org/web/20111007231611/http://www.axiaaudio.com/tech/voip/default.htm). Axia Audio. Archived from the original (http://www.axiaaudio.com/tech/voip/default.htm) on October 7, 2011. Retrieved June 21, 2011.
10. Jackson, William (May 27, 2009). "SSA goes big on VOIP" (http://gcn.com/Articles/2009/06/01/SSA-VOIP-implementation.aspx?s=gcndaily_280509&Page=1). Government Computer News. Retrieved 2009-05-28.
11. "Social Security to Build "World's Largest VOIP" " (http://www.govtech.com/gt/275677). Government Technology. Retrieved 2009-05-29.
12. Korzeniowski, Peter (January 8, 2009). "Three Technologies You Need In 2009" (https://www.forbes.com/2009/01/08/small-business-voip-ent-tech-cx_bm_0108bmightytech09.html/). *Forbes*. Retrieved 2009-03-02.
13. Callahan, Renee (December 9, 2008). "Businesses Move To Voice-Over-IP" (https://www.forbes.com/2008/12/09/skype-vonage-ringcentral_leadership_clayton_in_rc_1209claytonchristensen_inl.html). *Forbes*. Retrieved 2009-03-03.
14. "Skype For Business" (http://www.skype.com/business/allfeatures/3skypephone/). skype.com. Retrieved 2009-03-16.
15. "VoIP– Vulnerability over Internet Protocol?" (http://www.continuitycentral.com/feature074.htm).
16. "Quality of Service for Voice over IP" (http://www.cisco.com/en/US/docs/ios/solutions_docs/qos_solutions/QoSVoIP/QoSVoIP.html). Retrieved May 3, 2011.
17. Prabhakar, G.; Rastogi, R.; Thotton, M (2005). "OSS Architecture & Requirements for VoIP Networks". *Bell Labs Technical Journal*. **10** (1): 31–45. doi:10.1002/bltj.20077 (https://doi.org/10.1002%2Fbltj.20077).
18. "Quality of Service for Voice over IP" (http://www.cisco.com/en/US/docs/ios/solutions_docs/qos_solutions/QoSVoIP/QoSVoIP.html#wp1029054). Retrieved May 3, 2011.
19. Li, Hong; Mason, Lorne (28–30 April 2008). *IEEE Multipath routing with adaptive playback scheduling for Voice over IP in Service Overlay Networks. Sarnoff Symposium, 2008 IEEE*. pp. 1–5. CiteSeerX 10.1.1.214.7566 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.214.7566). doi:10.1109/SARNOF.2008.4520089 (https://doi.org/10.1109%2FSARNOF.2008.4520089). ISBN 978-1-4244-1843-5.

20. Caceres, Ramon. "RFC 3611: RTP Control Protocol Extended Reports (RTCP XR)" (https://tools.ietf.org/html/rfc3611.html). *tools.ietf.org*. Retrieved 2019-04-12.
21. CableLabs, *PacketCable Residential SIP Telephony Feature Definition*, Technical Report, PKT-TR-RST-V03-071106 (2007)
22. "VoIP performance measurement using QoS parameters" (http://www.it-innovations.ae/iit005/proceedings/articles/H_3_IIT05_Amin.pdf) (PDF). A.H.Muhamad Amin. 2016-08-14.
23. "Methodology for SIP Infrastructure Performance Testing" (http://www.wseas.us/e-library/transactions/computers/2010/88-134.pdf) (PDF). Miroslav Voznak, Jan Rozhon. 2016-08-14.
24. "Voice over IP (VoIP) Performance Evaluation on VMware vSphere® 5" (http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/voip-performance-vsphere5-white-paper.pdf) (PDF). VMware. 2016-08-14.
25. "Performance and Stress Testing of SIP Servers, Clients and IP Networks" (http://startrinity.com/VoIP/TestingSipPbxSoftswitchServer.aspx). StarTrinity. 2016-08-13.
26. "Testing Voice over IP (VoIP) Networks" (https://www.ixiacom.com/sites/default/files/resources/whitepaper/voip-whitepaper.pdf) (PDF). IXIA. 2016-08-14.
27. "Importance of Softswitch VoIP Technology" (http://www.ixc.ua/importance-of-softswitch-voip-technology-properly). ixc.ua. May 20, 2011. Retrieved 2012-10-04.
28. "RFC 3824– Using E.164 numbers with the Session Initiation Protocol (SIP)" (http://www.packetizer.com/rfc/rfc3824/). The Internet Society. June 1, 2004. Retrieved 2009-01-21.
29. "Create a Skype Name" (http://www.skype.com/help/guides/createskypename_windows/). Skype. Retrieved 2009-01-21.
30. "RFC 3969– The Internet Assigned Number Authority (IANA) Uniform Resource Identifier (URI) Parameter Registry for the Session Initiation Protocol (SIP)" (http://www.packetizer.com/rfc/rfc3969/). The Internet Society. December 1, 2004. Retrieved 2009-01-21.
31. "Your personal online number" (http://www.skype.com/allfeatures/onlinenumber/). Skype. Retrieved 2009-01-21.
32. "Application-level Network Interoperability and the Evolution of IMS" (http://ipcommunications.tmcnet.com/hot-topics/MCP/articles/1311-application-level-network-interoperability-the-evolution-ims.htm). TMCnet.com. May 24, 2006. Retrieved 2009-01-21.
33. Jeff Riddel (2007). *Packetcable Implementation* (https://books.google.com/books?id=8CNBbrxytcAC&pg=PA557). Cisco Press. p. 557. ISBN 978-1-58705-181-4.
34. "Keeping your telephone number when you change your service provider" (http://www.fcc.gov/cgb/consumerfacts/numbport.html). *FCC*.
35. "FCC Consumer Advisory VoIP and 911 Service" (http://www.fcc.gov/cgb/consumerfacts/voip911.pdf) (PDF). Retrieved May 2, 2011.
36. Soft-Switch.org (http://soft-switch.org/foip.html), Faxing over IP networks
37. "UMass Discussion on UDP transmission Characteristics" (http://www-net.cs.umass.edu/kurose/transport/UDP.html).
38. "ICT Regulation Tool Kit – 4.4 VOIP – Regulatory Issues – Universal Service" (http://www.ictregulationtoolkit.org/en/Section.3083.html). Retrieved September 21, 2017.
39. "Examining Two Well-Known Attacks on VoIP" (http://www.circleid.com/posts/examining_two_well_known_attacks_on_voip1/). *CircleID*. Retrieved 2006-04-05.
40. Disa.mil (http://iase.disa.mil/stigs/stig/VoIP-STIG-V2R2.pdf), Internet Protocol Telephony & Voice over Internet Protocol Security Technical Implementation Guide
41. Secure Voice over IP (SVoIP) vs. Voice over Secure IP (VoSIP) Installations (http://www.gdc4s.com/Documents/Products/SecureVoiceData/GD-SVOIP_FAQ-w.pdf) General Dynamics C4 Systems
42. Dunte, Markus; Ruland, Christoph (June 2007). "Secure Voice-over-IP" (http://paper.ijcsns.org/07_book/200706/20070610.pdf) (PDF). *International Journal of Computer Science and Network Security*. **7** (6): 63–68.
43. Sans.org (http://www.sans.org/reading_room/whitepapers/voip/secure_voice_over_ip_322), SANS Institute InfoSec Reading Room

44. White, C.M.; Teague, K.A.; Daniel, E.J. (7–10 Nov 2004). *Browse Conference Publications > Signals, Systems and Computer ... Help Working with Abstracts Packet loss concealment in a secure voice over IP environment* (http://www.clsp.jhu.edu/~cwhite/papers/asilo_04_LossConceal_final.pdf) (PDF). *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*. **1**. pp. 415–419. CiteSeerX 10.1.1.219.633 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.219.633). doi:10.1109/ACSSC.2004.1399165 (https://doi.org/10.1109%2FACSSC.2004.1399165). ISBN 978-0-7803-8622-8.
45. "Cellcrypt secure VOIP heading to BlackBerry" (http://www.networkworld.com/news/2009/041609-cellcrypt-secure-voip-heading-to.html). *Networkworld.com*.
46. "Secure VOIP calling, free software, and the right to privacy" (http://www.freesoftwaremagazine.com/columns/secure_voip_calling_free_software_right_to_privacy). *Free Software Magazine*.
47. VOIPSA.org (http://voipsa.org/blog/2006/09/29/hello-mom-im-a-fake/), Blog: "Hello Mom, I'm a Fake!" (Telespoof and Fakecaller).
48. FCC.gov (http://www.fcc.gov/voip/), What are some advantages of VoIP?
49. "Voip Infrastructure" (http://www.hp.com/rnd/pdfs/final_voip_techbrief.pdf) (PDF).
50. "Global VOIP Policy Status Matrix" (http://www.ipall.org/matrix/). Global IP Alliance. Retrieved 2006-11-23.
51. Proenza, Francisco J. "The Road to Broadband Development in Developing Countries is through Competition Driven by Wireless and VOIP" (http://www.e-forall.org/pdf/Wireless&VOIP_10July2006.pdf) (PDF). Retrieved 2008-04-07.
52. "Telecom Decision CRTC 2005-21" (http://crtc.gc.ca/eng/archive/2005/dt2005-21.htm). *Canadian Radio-television and Telecommunications Commission*. Government of Canada. 4 April 2005. Retrieved 29 April 2017.
53. Metz, Cade. "Oman cuffs 212 for selling VoIP calls" (https://www.theregister.co.uk/2009/11/20/oman_and_voip/). *The Register*. Retrieved 20 September 2016.
54. "Saudi Arabia to lift ban on internet calls" (https://www.bbc.com/news/world-middle-east-41332743). *BBC News*. 20 September 2017. Retrieved 10 January 2018.
55. "Saudi Arabia to lift ban on internet calls" (https://www.reuters.com/article/us-saudi-telecoms-ban/saudi-arabia-to-lift-ban-on-internet-calls-idUSKCN1BV128). *Reuters*. 20 September 2017. Retrieved 10 January 2018.
56. "Don't worry, Skype is working in UAE" (https://www.khaleejtimes.com/technology/dont-worry-skype-is-working-in-uae-). *Khaleejtimes*. 26 June 2017. Retrieved 11 January 2018.
57. Debusmann Jr, Bernd (9 January 2018). "Etisalat launches new unlimited calling plan with VoIP apps" (http://www.arabianbusiness.com/technology/387211-etisalat-launches-new-unlimited-calling-plan-with-voip-apps). *Arabian Business*. Retrieved 9 January 2018.
58. Maceda, Cleofe (8 January 2018). "No Skype? Pay Dh50 monthly for video calls" (http://gulfnews.com/business/sectors/technology/no-skype-pay-dh50-monthly-for-video-calls-1.2153623). *Gulf News*. Retrieved 9 January 2018.
59. Zacharias, Anna \ (8 January 2018). "Etisalat launches new calling app plan days after Skype disruptions" (https://www.thenational.ae/uae/etisalat-launches-new-calling-app-plan-days-after-skype-disruptions-1.693837). *The National*. Retrieved 9 January 2018.
60. Mahanagar Doorsanchar Bhawan and Jawahar Lal Nehru Marg (May 2008). "Telecom Regulatory Authority of India (TRAI) Consultation paper on Issues related to Internet Telephony. Consultation Paper No. 11/2008" (https://web.archive.org/web/20141006100940/http://www.trai.gov.in/WriteReaddata/ConsultationPaper/Document/cpaper12may08.pdf) (PDF). New Delhi India: Telecom Regulatory Authority of India (TRAI). p. 16 (Section 2.2.1.2 PC–to–Phone Internet telephony). Archived from the original (http://www.trai.gov.in/WriteReaddata/ConsultationPaper/Document/cpaper12may08.pdf) (PDF) on 2014-10-06. Retrieved September 19, 2012. "An end user is allowed to make PC–to-Phone Internet Telephony calls only on PSTN/PLMN abroad."
61. Harish Kumar Gangwar Technical Note on Illegal International Long Distance telephone Exchange in India (https://www.scribd.com/doc/101919043/TECHNICAL-NOTE-ON-ILLEGAL-INTERNATIONAL-LONG-DISTANCE-TELEPHONE-EXCHANGE-IN-INDIA)

62. Stripes.com (http://www.stripes.com/article.asp?section=104&article=41826&archive=true), Stars and Stripes: USFK deal keeps VoIP access for troops
63. Pershing, Genny. "Cybertelecom :: VoIP :: FCC" (http://www.cybertelecom.org/voip/fcc.htm). *www.cybertelecom.org*. Retrieved September 21, 2017.
64. GPO.gov (http://www.access.gpo.gov/nara/cfr/waisidx_07/47cfr9_07.html), 47 C.F.R. pt. 9 (2007)
65. "VoIP and 911 Service" (http://www.fcc.gov/guides/voip-and-911-service). FCC. Retrieved 16 August 2014.
66. "Voice Over Internet Protocol (VoIP)" (http://www.fcc.gov/voip/). November 18, 2010. Retrieved September 21, 2017.
67. Greenberg, Andy (May 15, 2008). "The State Of Cybersecurity Wiretapping's Fuzzy Future" (https://www.forbes.com/2008/05/15/wiretapping-voip-lichtblau-tech-security08-cx_ag_0515wiretap.html/). *Forbes*. Retrieved 2009-03-02.
68. "Danny Cohen" (http://www.internethalloffame.org/inductees/danny-cohen). INTERNET HALL of FAME. Retrieved 2014-12-06.
69. *Advanced Content Delivery, Streaming, and Cloud Services (Pg 34)* (https://books.google.com/?id=3yaUBAAAQBAJ&pg=PA34&lpg=PA34&dq=Network+Voice+Protocol+%28NVP%29+developed+by+Danny+Cohen). Willey. 2014-09-19. ISBN 9781118909706. Retrieved 2014-12-06.
70. Cerf, V.; Kahn, R. (May 1974). "A Protocol for Packet Network Intercommunication" (http://www.cs.rice.edu/~eugeneng/teaching/f07/comp529/papers/ck74.pdf) (PDF). *IEEE Transactions on Communications*. **22** (5): 637–648. doi:10.1109/TCOM.1974.1092259 (https://doi.org/10.1109%2FTCOM.1974.1092259).
71. "The Launch of NSFNET" (https://www.nsf.gov/about/history/nsf0050/internet/launch.htm). The National Science Foundation. Retrieved 2009-01-21.
72. "Speak Freely History" (http://www.speakfreely.org/history.html). Brian C. Wiles. April 18, 1999. Retrieved 2013-03-19.
73. IDG Network World Inc; Eckerson, Wayne (21 September 1992). *Network World - Startup targets desktop Videoconferencing arena* (https://books.google.com/books?id=DhQEAAAAMBAJ&pg=PA39). IDG Network World Inc. pp. 39–. ISSN 0887-7661 (https://www.worldcat.org/issn/0887-7661). Retrieved 10 February 2012.
74. "Executive Profile, Dan Harple" (http://investing.businessweek.com/research/stocks/private/person.asp?personId=432652&privcapId=33018365&previousCapId=3569924&previousTitle=Slater%20Center%20for%20Interactive%20Technologies). *Bloomberg Businessweek*. Bloomberg.com. Retrieved 20 December 2018.
75. "Company Overview" (http://investing.businessweek.com/research/stocks/private/snapshot.asp?privcapId=29980). *Software: InSoft, Inc. February 10, 2012*. Bloomberg Businessweek. Retrieved 20 December 2018.
76. "MTALK-Readme" (ftp://sunsite.unc.edu/pub/Linux/apps/sound/talk/mtalk.README) (TXT). Sunsite.edu. Retrieved 2012-04-29.
77. Keating, Tom. "Internet Phone Release 4" (http://blog.tmcnet.com/blog/tom-keating/docs/cti-buyers-guide-1996.pdf) (PDF). Computer Telephony Interaction Magazine. Retrieved 2007-11-07.
78. "The 10 that Established VOIP (Part 1: VocalTec)" (http://www.ilocus.com/2007/07/the_10_that_established_voip_p_2.html). iLocus. Retrieved 2009-01-21.
79. The free Library RADVision and Intel Target Compatibility Between RADVision's H.323/320 Videoconferencing Gateway And Intel's Business Video Conferencing And TeamStation Products. (http://www.thefreelibrary.com/RADVision+and+Intel+Target+Compatibility+Between+RADVision%27s+...-a019467970) June 2, 1997 VoiP Developer Solutions (http://www.radvision.com/Developer-Solutions/)
80. "H.323 Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service" (http://www.itu.int/rec/T-REC-H.323-199611-S/en). ITU-T. Retrieved 2009-01-21.
81. "RFC 2235" (http://www.faqs.org/rfcs/rfc2235.html). R. Zakon. Retrieved 2009-01-21.
82. "The 10 that Established VOIP (Part 2: Level 3)" (http://www.ilocus.com/2007/07/the_10_that_established_voip_p_1.html). iLocus. July 13, 2007. Retrieved 2007-11-07.
83. "RFC 2543, SIP: Session Initiation Protocol" (http://www.ietf.org/rfc/rfc2543.txt). Handley, Schulzrinne, Schooler, Rosenberg. Retrieved 2009-01-21.

84. "What is Asterisk" (http://www.asterisk.org/about). Asterisk.org. Retrieved 2009-01-21.
85. Remo, Michelle V. (August 27, 2007). "Prospects bright for voice calls over internet" (https://news.google.com/newspapers?nid=2479&dat=20070827&id=j1M1AAAAIBAJ&sjid=YyUMAAAAIBAJ&pg=1974,4860651). *Philippine Daily Inquirer*. Retrieved 2015-01-01.

External links

- The dictionary definition of *VoIP* at Wiktionary
- Internet telephony travel guide from Wikivoyage

WIKIPEDIA

File Transfer Protocol

The **File Transfer Protocol** (**FTP**) is a standard network protocol used for the transfer of computer files between a client and server on a computer network.

FTP is built on a client-server model architecture using separate control and data connections between the client and the server.[1] FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS) or replaced with SSH File Transfer Protocol (SFTP).

The first FTP client applications were command-line programs developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems.[2][3] Many FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as HTML editors.

Contents

History of FTP servers

The original specification for the File Transfer Protocol was written by Abhay Bhushan and published as RFC 114 (https://tools.ietf.org/html/rfc114) on 16 April 1971. Until 1980, FTP ran on NCP, the predecessor of TCP/IP.[2] The protocol was later replaced by a TCP/IP version, RFC 765 (https://tools.ietf.org/html/rfc765) (June 1980) and RFC 959 (https://tools.ietf.org/html/rfc959) (October 1985), the current specification. Several proposed standards amend RFC 959 (https://tools.ietf.org/html/rfc959), for example RFC 1579 (https://tools.ietf.org/html/rfc1579) (February 1994) enables Firewall-Friendly FTP (passive mode), RFC 2228 (https://tools.ietf.org/html/rfc2228) (June 1997) proposes security extensions, RFC 2428 (https://tools.ietf.org/html/rfc2428) (September 1998) adds support for IPv6 and defines a new type of passive mode.[4]

Protocol overview

Communication and data transfer

FTP may run in *active* or *passive* mode, which determines how the data connection is established.[5] In both cases, the client creates a TCP control connection from a random, usually an unprivileged, port N to the FTP server command port 21.

- In active mode, the client starts listening for incoming data connections from the server on port M. It sends the FTP command PORT M to inform the server on which port it is listening. The server then initiates a data channel to the client from its port 20, the FTP server data port.
- In situations where the client is behind a firewall and unable to accept incoming TCP connections, *passive mode* may be used. In this mode, the client uses the control connection to send a PASV command to the server and then receives a server IP address and server port number from the server,[5] which the client then uses to open a data connection from an arbitrary client port to the server IP address and server port number received.[6]

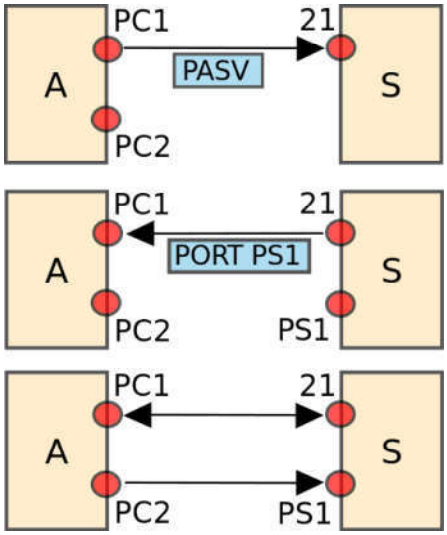


Illustration of starting a passive connection using port 21

Both modes were updated in September 1998 to support IPv6. Further changes were introduced to the passive mode at that time, updating it to *extended passive mode*.[7]

The server responds over the control connection with three-digit status codes in ASCII with an optional text message. For example, "200" (or "200 OK") means that the last command was successful. The numbers represent the code for the response and the optional text represents a human-readable explanation or request (e.g. <Need account for storing file>).[1] An ongoing transfer of file data over the data connection can be aborted using an interrupt message sent over the control connection.

While transferring data over the network, four data representations can be used:[2][3][4]

- ASCII mode: Used for text. Data is converted, if needed, from the sending host's character representation to "8-bit ASCII" before transmission, and (again, if necessary) to the receiving host's character representation. As a consequence, this mode is inappropriate for files that contain data other than plain text.
- Image mode (commonly called Binary mode): The sending machine sends each file byte by byte, and the recipient stores the bytestream as it receives it. (Image mode support has been recommended for all implementations of FTP).
- EBCDIC mode: Used for plain text between hosts using the EBCDIC character set.
- Local mode: Allows two computers with identical setups to send data in a proprietary format without the need to convert it to ASCII.

For text files, different format control and record structure options are provided. These features were designed to facilitate files containing Telnet or ASA.

Data transfer can be done in any of three modes:[1][2]

- Stream mode: Data is sent as a continuous stream, relieving FTP from doing any processing. Rather, all processing is left up to TCP. No End-of-file indicator is needed, unless the data is divided into records.
- Block mode: FTP breaks the data into several blocks (block header, byte count, and data field) and then passes it on to TCP.[4]
- Compressed mode: Data is compressed using a simple algorithm (usually run-length encoding).

Some FTP software also implements a DEFLATE-based compressed mode, sometimes called "Mode Z" after the command that enables it. This mode was described in an Internet Draft, but not standardized.[8]

Login

FTP login uses normal username and password scheme for granting access.[2] The username is sent to the server using the USER command, and the password is sent using the PASS command.[2] This sequence is unencrypted "on the wire", so may be vulnerable to a network sniffing attack.[9] If the information provided by the client is accepted by the server, the server will send a greeting to the client and the session will commence.[2] If the server supports it, users may log in without providing login credentials, but the same server may authorize only limited access for such sessions.[2]

Anonymous FTP

A host that provides an FTP service may provide anonymous FTP access.[2] Users typically log into the service with an 'anonymous' (lower-case and case-sensitive in some FTP servers) account when prompted for user name. Although users are commonly asked to send their email address instead of a password,[3] no verification is actually performed on the supplied data.[10] Many FTP hosts whose purpose is to provide software updates will allow anonymous logins.[3]

NAT and firewall traversal

FTP normally transfers data by having the server connect back to the client, after the PORT command is sent by the client. This is problematic for both NATs and firewalls, which do not allow connections from the Internet towards internal hosts.[11] For NATs, an additional complication is that the representation of the IP addresses and port number in the PORT command refer to the internal host's IP address and port, rather than the public IP address and port of the NAT.

There are two approaches to solve this problem. One is that the FTP client and FTP server use the PASV command, which causes the data connection to be established from the FTP client to the server.[11] This is widely used by modern FTP clients. Another approach is for the NAT to alter the values of the PORT command, using an application-level gateway for this purpose.[11]

Differences from HTTP

HTTP essentially fixes the bugs in FTP that made it inconvenient to use for many small ephemeral transfers as are typical in web pages.

FTP has a stateful control connection which maintains a current working directory and other flags, and each transfer requires a secondary connection through which the data are transferred. In "passive" mode this secondary connection is from client to server, whereas in the default "active" mode this connection is from server to client. This apparent role

reversal when in active mode, and random port numbers for all transfers, is why firewalls and NAT gateways have such a hard time with FTP. HTTP is stateless and multiplexes control and data over a single connection from client to server on well-known port numbers, which trivially passes through NAT gateways and is simple for firewalls to manage.

Setting up an FTP control connection is quite slow due to the round-trip delays of sending all of the required commands and awaiting responses, so it is customary to bring up a control connection and hold it open for multiple file transfers rather than drop and re-establish the session afresh each time. In contrast, HTTP originally dropped the connection after each transfer because doing so was so cheap. While HTTP has subsequently gained the ability to reuse the TCP connection for multiple transfers, the conceptual model is still of independent requests rather than a session.

When FTP is transferring over the data connection, the control connection is idle. If the transfer takes too long, the firewall or NAT may decide that the control connection is dead and stop tracking it, effectively breaking the connection and confusing the download. The single HTTP connection is only idle between requests and it is normal and expected for such connections to be dropped after a time-out.

Web browser support

Most common web browsers can retrieve files hosted on FTP servers, although they may not support protocol extensions such as FTPS.[3][12] When an FTP—rather than an HTTP—URL is supplied, the accessible contents on the remote server are presented in a manner that is similar to that used for other web content. A full-featured FTP client can be run within Firefox in the form of an extension called FireFTP.

Syntax

FTP URL syntax is described in RFC 1738 (https://tools.ietf.org/html/rfc1738), taking the form: `ftp://[user[:password]@]host[:port]/url-path` (the bracketed parts are optional).

For example, the URL ftp://public.ftp-servers.example.com/mydirectory/myfile.txt represents the file *myfile.txt* from the directory *mydirectory* on the server *public.ftp-servers.example.com* as an FTP resource. The URL ftp://user001:secretpassword@private.ftp-servers.example.com/mydirectory/myfile.txt adds a specification of the username and password that must be used to access this resource.

More details on specifying a username and password may be found in the browsers' documentation (e.g., Firefox[13] and Internet Explorer[14]). By default, most web browsers use passive (PASV) mode, which more easily traverses end-user firewalls.

Some variation has existed in how different browsers treat path resolution in cases where there is a non-root home directory for a user.[15]

Security

FTP was not designed to be a secure protocol, and has many security weaknesses.[16] In May 1999, the authors of RFC 2577 (https://tools.ietf.org/html/rfc2577) listed a vulnerability to the following problems:

- Brute force attack
- FTP bounce attack
- Packet capture
- Port stealing (guessing the next open port and usurping a legitimate connection)
- Spoofing attack
- Username enumeration

FTP does not encrypt its traffic; all transmissions are in clear text, and usernames, passwords, commands and data can be read by anyone able to perform packet capture (sniffing) on the network.[2][16] This problem is common to many of the Internet Protocol specifications (such as SMTP, Telnet, POP and IMAP) that were designed prior to the creation of encryption mechanisms such as TLS or SSL.[4]

Common solutions to this problem include:

1. Using the secure versions of the insecure protocols, e.g., FTPS instead of FTP and TelnetS instead of Telnet.
2. Using a different, more secure protocol that can handle the job, e.g. SSH File Transfer Protocol or Secure Copy Protocol.
3. Using a secure tunnel such as Secure Shell (SSH) or virtual private network (VPN).

FTP over SSH

FTP over SSH is the practice of tunneling a normal FTP session over a Secure Shell connection.[16] Because FTP uses multiple TCP connections (unusual for a TCP/IP protocol that is still in use), it is particularly difficult to tunnel over SSH. With many SSH clients, attempting to set up a tunnel for the control channel (the initial client-to-server connection on port 21) will protect only that channel; when data is transferred, the FTP software at either end sets up new TCP connections (data channels) and thus have no confidentiality or integrity protection.

Otherwise, it is necessary for the SSH client software to have specific knowledge of the FTP protocol, to monitor and rewrite FTP control channel messages and autonomously open new packet forwardings for FTP data channels. Software packages that support this mode include:

- Tectia ConnectSecure (Win/Linux/Unix)[17] of SSH Communications Security's software suite

Derivatives

FTPS

Explicit FTPS is an extension to the FTP standard that allows clients to request FTP sessions to be encrypted. This is done by sending the "AUTH TLS" command. The server has the option of allowing or denying connections that do not request TLS. This protocol extension is defined in RFC 4217 (https://tools.ietf.org/html/rfc4217). Implicit FTPS is an outdated standard for FTP that required the use of a SSL or TLS connection. It was specified to use different ports than plain FTP.

SSH File Transfer Protocol

The SSH file transfer protocol (chronologically the second of the two protocols abbreviated SFTP) transfers files and has a similar command set for users, but uses the Secure Shell protocol (SSH) to transfer files. Unlike FTP, it encrypts both commands and data, preventing passwords and sensitive information from being transmitted openly over the network. It cannot interoperate with FTP software.

Trivial File Transfer Protocol

Trivial File Transfer Protocol (TFTP) is a simple, lock-step FTP that allows a client to get a file from or put a file onto a remote host. One of its primary uses is in the early stages of booting from a local area network, because TFTP is very simple to implement. TFTP lacks security and most of the advanced features offered by more robust file transfer protocols such as File Transfer Protocol. TFTP was first standardized in 1981 and the current specification for the protocol can be found in RFC 1350 (https://tools.ietf.org/html/rfc1350).

Simple File Transfer Protocol

Simple File Transfer Protocol (the first protocol abbreviated SFTP), as defined by RFC 913 (https://tools.ietf.org/html/rfc913), was proposed as an (unsecured) file transfer protocol with a level of complexity intermediate between TFTP and FTP. It was never widely accepted on the Internet, and is now assigned Historic status by the IETF. It runs through port 115, and often receives the initialism of *SFTP*. It has a command set of 11 commands and support three types of data transmission: ASCII, binary and continuous. For systems with a word size that is a multiple of 8 bits, the implementation of binary and continuous is the same. The protocol also supports login with user ID and password, hierarchical folders and file management (including *rename*, *delete*, *upload*, *download*, *download with overwrite*, and *download with append*).

FTP commands

FTP reply codes

Below is a summary of FTP reply codes that may be returned by an FTP server. These codes have been standardized in RFC 959 (https://tools.ietf.org/html/rfc959) by the IETF. The reply code is a three-digit value. The first digit is used to indicate one of three possible outcomes — success, failure, or to indicate an error or incomplete reply:

- 2yz – Success reply
- 4yz or 5yz – Failure reply
- 1yz or 3yz – Error or Incomplete reply

The second digit defines the kind of error:

- x0z – Syntax. These replies refer to syntax errors.
- x1z – Information. Replies to requests for information.
- x2z – Connections. Replies referring to the control and data connections.
- x3z – Authentication and accounting. Replies for the login process and accounting procedures.
- x4z – Not defined.
- x5z – File system. These replies relay status codes from the server file system.

The third digit of the reply code is used to provide additional detail for each of the categories defined by the second digit.

FTP servers

Some popular open source and commercial FTP server implementations are:

- FileZilla Server (Windows)
- Pure-FTPd (Unix)
- Vsftpd (Unix)
- ProFTPd (Unix)

See also

- Comparison of FTP client software
- Comparison of FTP server software
- Comparison of file transfer protocols
- Curl-loader – FTP/S loading/testing open-source software
- File eXchange Protocol (FXP)
- File Service Protocol (FSP)
- FTAM
- FTPFS
- List of FTP commands
- List of FTP server return codes
- List of FTP server software
- Managed File Transfer

- OBEX
- Shared file access
- TCP Wrapper

References

1. Forouzan, B.A. (2000). *TCP/IP: Protocol Suite* (1st ed.). New Delhi, India: Tata McGraw-Hill Publishing Company Limited.
2. Kozierok, Charles M. (2005). "The TCP/IP Guide v3.0" (http://www.tcpipguide.com/free/t_FTPOverviewHistoryandStandards.htm). Tcpipguide.com.
3. Dean, Tamara (2010). *Network+ Guide to Networks*. Delmar. pp. 168–171.
4. Clark, M.P. (2003). *Data Networks IP and the Internet* (1st ed.). West Sussex, England: John Wiley & Sons Ltd.
5. "Active FTP vs. Passive FTP, a Definitive Explanation" (https://web.archive.org/web/20110504071617/http://slacksite.com/other/ftp.html). Slacksite.com. Archived from the original (http://slacksite.com/other/ftp.html) on 4 May 2011.
6. RFC 959 (https://tools.ietf.org/html/rfc959) (Standard) File Transfer Protocol (FTP). Postel, J. & Reynolds, J. (October 1985).
7. RFC 2428 (https://tools.ietf.org/html/rfc2428) (Proposed Standard) Extensions for IPv6, NAT, and Extended Passive Mode. Allman, M. & Metz, C. & Ostermann, S. (September 1998).
8. Preston, J. (January 2005). *Deflate transmission mode for FTP* (https://tools.ietf.org/html/draft-preston-ftpext-deflate-03.txt). IETF. I-D draft-preston-ftpext-deflate-03.txt. Retrieved 27 January 2016.
9. Prince, Brian. "Should Organizations Retire FTP for Security?" (http://www.securityweek.com/should-organizations-retire-ftp-security). *Security Week*. Security Week. Retrieved 14 September 2017.
10. RFC 1635 (https://tools.ietf.org/html/rfc1635) (Informational) How to Use Anonymous FTP. P. & Emtage, A. & Marine, A. (May 1994).
11. Gleason, Mike (2005). "The File Transfer Protocol and Your Firewall/NAT" (http://www.ncftp.com/ncftpd/doc/misc/ftp_and_firewalls.html). Ncftp.com.
12. Matthews, J. (2005). *Computer Networking: Internet Protocols in Action* (1st ed.). Danvers, MA: John Wiley & Sons Inc.
13. "Accessing FTP servers | How to | Firefox Help" (http://support.mozilla.com/en-US/kb/Accessing+FTP+servers#FTP_servers_that_require_a_username_and_password). Support.mozilla.com. 2012-09-05. Retrieved 2013-01-16.
14. "How to Enter FTP Site Password in Internet Explorer" (http://support.microsoft.com/kb/135975). Support.microsoft.com. 2011-09-23. Retrieved 2015-03-28. Written for IE versions 6 and earlier. Might work with newer versions.
15. Jukka "Yucca" Korpela (1997-09-18). "FTP URLs" (https://www.cs.tut.fi/~jkorpela/ftpurl.html). "IT and communication" (www.cs.tut.fi/~jkorpela/). Retrieved 2016-01-06.
16. "Securing FTP using SSH" (http://www.nurdletech.com/linux-notes/ftp/ssh.html). Nurdletech.com.
17. "Access using SSH keys & PCI DSS compliance" (http://ssh.com/index.php/products/tectia-pci-point-to-point-encryption.html). *ssh.com*.

Further reading

- RFC 697 (https://tools.ietf.org/html/rfc697) – CWD Command of FTP. July 1975.
- RFC 959 (https://tools.ietf.org/html/rfc959) – (Standard) File Transfer Protocol (FTP). J. Postel, J. Reynolds. October 1985.
- RFC 1579 (https://tools.ietf.org/html/rfc1579) – (Informational) Firewall-Friendly FTP. February 1994.
- RFC 1635 (https://tools.ietf.org/html/rfc1635) – (Informational) How to Use Anonymous FTP. May 1994.
- RFC 1639 (https://tools.ietf.org/html/rfc1639) – FTP Operation Over Big Address Records (FOOBAR). June 1994.
- RFC 1738 (https://tools.ietf.org/html/rfc1738) – Uniform Resource Locators (URL). December 1994.

- RFC 2228 (https://tools.ietf.org/html/rfc2228) – (Proposed Standard) FTP Security Extensions. October 1997.
- RFC 2389 (https://tools.ietf.org/html/rfc2389) – (Proposed Standard) Feature negotiation mechanism for the File Transfer Protocol. August 1998.
- RFC 2428 (https://tools.ietf.org/html/rfc2428) – (Proposed Standard) Extensions for IPv6, NAT, and Extended passive mode. September 1998.
- RFC 2577 (https://tools.ietf.org/html/rfc2577) – (Informational) FTP Security Considerations. May 1999.
- RFC 2640 (https://tools.ietf.org/html/rfc2640) – (Proposed Standard) Internationalization of the File Transfer Protocol. July 1999.
- RFC 3659 (https://tools.ietf.org/html/rfc3659) – (Proposed Standard) Extensions to FTP. P. Hethmon. March 2007.
- RFC 5797 (https://tools.ietf.org/html/rfc5797) – (Proposed Standard) FTP Command and Extension Registry. March 2010.
- RFC 7151 (https://tools.ietf.org/html/rfc7151) – (Proposed Standard) File Transfer Protocol HOST Command for Virtual Hosts. March 2014.
- IANA FTP Commands and Extensions registry (http://www.iana.org/assignments/ftp-commands-extensions/ftp-commands-extensions.xhtml) – The official registry of FTP Commands and Extensions

External links

- Communication Networks/File Transfer Protocol at Wikibooks
- FTP Server Online Tester (https://servertest.online/ftp) Authentication, encryption, mode and connectivity.

WIKIPEDIA

Domain Name System

The **Domain Name System** (**DNS**) is a hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates more readily memorized domain names to the numerical IP addresses needed for locating and identifying computer services and devices with the underlying network protocols. By providing a worldwide, distributed directory service, the Domain Name System has been an essential component of the functionality of the Internet since 1985.

The Domain Name System delegates the responsibility of assigning domain names and mapping those names to Internet resources by designating authoritative name servers for each domain. Network administrators may delegate authority over sub-domains of their allocated name space to other name servers. This mechanism provides distributed and fault-tolerant service and was designed to avoid a single large central database.

The Domain Name System also specifies the technical functionality of the database service that is at its core. It defines the DNS protocol, a detailed specification of the data structures and data communication exchanges used in the DNS, as part of the Internet Protocol Suite.

The Internet maintains two principal namespaces, the domain name hierarchy[1] and the Internet Protocol (IP) address spaces.[2] The Domain Name System maintains the domain name hierarchy and provides translation services between it and the address spaces. Internet name servers and a communication protocol implement the Domain Name System.[3] A DNS name server is a server that stores the DNS records for a domain; a DNS name server responds with answers to queries against its database.

The most common types of records stored in the DNS database are for Start of Authority (SOA), IP addresses (A and AAAA), SMTP mail exchangers (MX), name servers (NS), pointers for reverse DNS lookups (PTR), and domain name aliases (CNAME). Although not intended to be a general purpose database, DNS has been expanded over time to store records for other types of data for either automatic lookups, such as DNSSEC records, or for human queries such as *responsible person* (RP) records. As a general purpose database, the DNS has also been used in combating unsolicited email (spam) by storing a real-time blackhole list (RBL). The DNS database is traditionally stored in a structured text file, the zone file, but other database systems are common.

Contents

Function

An often-used analogy to explain the Domain Name System is that it serves as the phone book for the Internet by translating human-friendly computer hostnames into IP addresses. For example, the domain name www.example.com translates to the addresses *93.184.216.34* (IPv4) and *2606:2800:220:1:248:1893:25c8:1946* (IPv6). The DNS can be quickly and transparently updated, allowing a service's location on the network to change without affecting the end users, who continue to use the same hostname. Users take advantage of this when they use meaningful Uniform Resource Locators (URLs), and e-mail addresses without having to know how the computer actually locates the services.

An important and ubiquitous function of DNS is its central role in distributed Internet services such as cloud services and content delivery networks.[4] When a user accesses a distributed Internet service using a URL, the domain name of the URL is translated to the IP address of a server that is proximal to the user. The key functionality of DNS exploited here is that different users can *simultaneously* receive different translations for the *same* domain name, a key point of divergence from a traditional phone-book view of the DNS. This process of using the DNS to assign proximal servers to users is key to providing faster and more reliable responses on the Internet and is widely used by most major Internet services.[5]

The DNS reflects the structure of administrative responsibility in the Internet.[6] Each subdomain is a zone of administrative autonomy delegated to a manager. For zones operated by a registry, administrative information is often complemented by the registry's RDAP and WHOIS services. That data can be used to gain insight on, and track

responsibility for, a given host on the Internet.[7]

History

Using a simpler, more memorable name in place of a host's numerical address dates back to the ARPANET era. The Stanford Research Institute (now SRI International) maintained a text file named HOSTS.TXT that mapped host names to the numerical addresses of computers on the ARPANET.[8][9] Elizabeth Feinler developed and maintained the first ARPANET directory.[10][11] Maintenance of numerical addresses, called the Assigned Numbers List, was handled by Jon Postel at the University of Southern California's Information Sciences Institute (ISI), whose team worked closely with SRI.[12]

Addresses were assigned manually. Computers, including their hostnames and addresses, were added to the master file by contacting the SRI's Network Information Center (NIC), directed by Elizabeth Feinler, by telephone during business hours.[13] Later, Feinler set up a WHOIS directory on a server in the NIC for retrieval of information about resources, contacts, and entities.[14] She and her team developed the concept of domains.[14] Feinler suggested that domains should be based on the location of the physical address of the computer.[15] Computers at educational institutions would have the domain *edu*, for example.[16] She and her team managed the Host Naming Registry from 1972 to 1989.[17]

By the early 1980s, maintaining a single, centralized host table had become slow and unwieldy and the emerging network required an automated naming system to address technical and personnel issues. Postel directed the task of forging a compromise between five competing proposals of solutions to Paul Mockapetris. Mockapetris instead created the Domain Name System.[13]

The Internet Engineering Task Force published the original specifications in RFC 882 and RFC 883 in November 1983.[18][19]

In 1984, four UC Berkeley students, Douglas Terry, Mark Painter, David Riggle, and Songnian Zhou, wrote the first Unix name server implementation for the Berkeley Internet Name Domain, commonly referred to as BIND.[20] In 1985, Kevin Dunlap of DEC substantially revised the DNS implementation. Mike Karels, Phil Almquist, and Paul Vixie have maintained BIND since then.[21] In the early 1990s, BIND was ported to the Windows NT platform. It was widely distributed, especially on Unix systems, and is still the most widely used DNS software on the Internet.[21]

In November 1987, RFC 1034[1] and RFC 1035[3] superseded the 1983 DNS specifications. Several additional Request for Comments have proposed extensions to the core DNS protocols.[22]

Structure

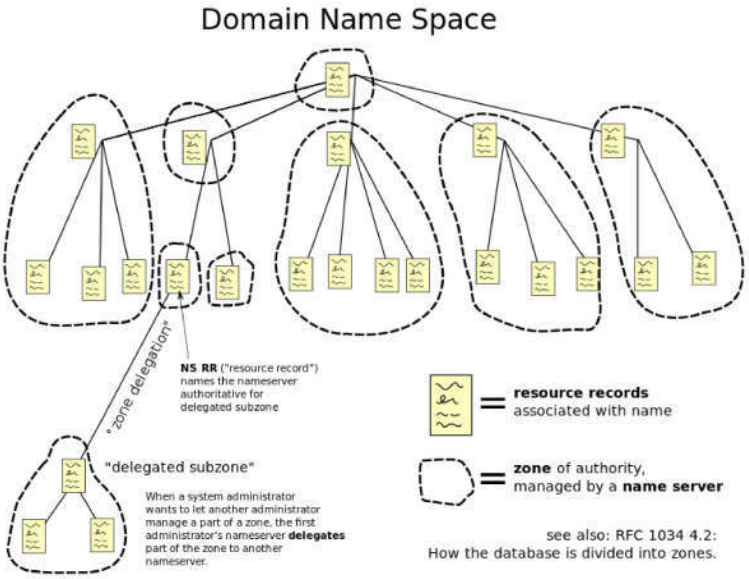
Domain name space

The domain name space consists of a tree data structure. Each node or leaf in the tree has a *label* and zero or more *resource records* (RR), which hold information associated with the domain name. The domain name itself consists of the label, possibly concatenated with the name of its parent node on the right, separated by a dot.[23]

The tree sub-divides into *zones* beginning at the root zone. A DNS zone may consist of only one domain, or may consist of many domains and sub-domains, depending on the administrative choices of the zone manager. DNS can also be partitioned according to *class* where the separate classes can be thought of as an array of parallel namespace trees.[24]

Administrative responsibility for any zone may be divided by creating additional zones. Authority over the new zone is said to be *delegated* to a designated name server. The parent zone ceases to be authoritative for the new zone.[24]

Domain name syntax, internationalization



The hierarchical Domain Name System for class *Internet*, organized into zones, each served by a name server

The definitive descriptions of the rules for forming domain names appear in RFC 1035, RFC 1123, RFC 2181, and RFC 5892. A domain name consists of one or more parts, technically called *labels*, that are conventionally concatenated, and delimited by dots, such as example.com.

The right-most label conveys the top-level domain; for example, the domain name www.example.com belongs to the top-level domain *com*.

The hierarchy of domains descends from right to left; each label to the left specifies a subdivision, or subdomain of the domain to the right. For example, the label *example* specifies a subdomain of the *com* domain, and *www* is a subdomain of example.com. This tree of subdivisions may have up to 127 levels.[25]

A label may contain zero to 63 characters. The null label, of length zero, is reserved for the root zone. The full domain name may not exceed the length of 253 characters in its textual representation.[1] In the internal binary representation of the DNS the maximum length requires 255 octets of storage, as it also stores the length of the name.[3]

Although no technical limitation exists to use any character in domain name labels which are representable by an octet, hostnames use a preferred format and character set. The characters allowed in labels are a subset of the ASCII character set, consisting of characters *a* through *z*, *A* through *Z*, digits *0* through *9*, and hyphen. This rule is known as the *LDH rule* (letters, digits, hyphen). Domain names are interpreted in case-independent manner.[26] Labels may not start or end with a hyphen.[27] An additional rule requires that top-level domain names should not be all-numeric.[27]

The limited set of ASCII characters permitted in the DNS prevented the representation of names and words of many languages in their native alphabets or scripts. To make this possible, ICANN approved the Internationalizing Domain Names in Applications (IDNA) system, by which user applications, such as web browsers, map Unicode strings into the valid DNS character set using Punycode. In 2009 ICANN approved the installation of internationalized domain name country code top-level domains (*ccTLD*s). In addition, many registries of the existing top-level domain names (*TLD*s) have adopted the IDNA system, guided by RFC 5890, RFC 5891, RFC 5892, RFC 5893.

Name servers

The Domain Name System is maintained by a distributed database system, which uses the client–server model. The nodes of this database are the name servers. Each domain has at least one authoritative DNS server that publishes information about that domain and the name servers of any domains subordinate to it. The top of the hierarchy is served by the root name servers, the servers to query when looking up (*resolving*) a TLD.

Authoritative name server

An *authoritative* name server is a name server that only gives answers to DNS queries from data that has been configured by an original source, for example, the domain administrator or by dynamic DNS methods, in contrast to answers obtained via a query to another name server that only maintains a cache of data.

An authoritative name server can either be a *master* server or a *slave* server. A master server is a server that stores the original (*master*) copies of all zone records. A slave server uses a special automatic updating mechanism in the DNS protocol in communication with its master to maintain an identical copy of the master records.

Every DNS zone must be assigned a set of authoritative name servers. This set of servers is stored in the parent domain zone with name server (NS) records.

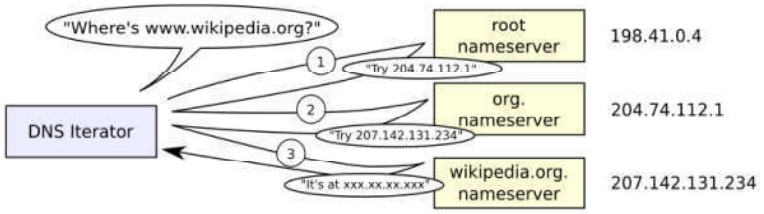
An authoritative server indicates its status of supplying definitive answers, deemed *authoritative*, by setting a protocol flag, called the "*Authoritative Answer*" (*AA*) bit in its responses.[3] This flag is usually reproduced prominently in the output of DNS administration query tools, such as dig, to indicate *that the responding name server is an authority for the domain name in question*.[3]

Operation

Address resolution mechanism

Domain name resolvers determine the domain name servers responsible for the domain name in question by a sequence of queries starting with the right-most (top-level) domain label.

For proper operation of its domain name resolver, a network host is configured with an initial cache (*hints*) of the known addresses of the root name servers. The hints are updated periodically by an administrator by retrieving a dataset from a reliable source.



A DNS resolver that implements the iterative approach mandated by RFC 1034; in this case, the resolver consults three name servers to resolve the fully qualified domain name "www.wikipedia.org".

Assuming the resolver has no cached records to accelerate the process, the resolution process starts with a query to one of the root servers. In typical operation, the root servers do not answer directly, but respond with a referral to more authoritative servers, e.g., a query for "www.wikipedia.org" is referred to the *org* servers. The resolver now queries the servers referred to, and iteratively repeats this process until it receives an authoritative answer. The diagram illustrates this process for the host that is named by the fully qualified domain name "www.wikipedia.org".

This mechanism would place a large traffic burden on the root servers, if every resolution on the Internet required starting at the root. In practice caching is used in DNS servers to off-load the root servers, and as a result, root name servers actually are involved in only a relatively small fraction of all requests.

Recursive and caching name server

In theory, authoritative name servers are sufficient for the operation of the Internet. However, with only authoritative name servers operating, every DNS query must start with recursive queries at the root zone of the Domain Name System and each user system would have to implement resolver software capable of recursive operation.

To improve efficiency, reduce DNS traffic across the Internet, and increase performance in end-user applications, the Domain Name System supports DNS cache servers which store DNS query results for a period of time determined in the configuration (*time-to-live*) of the domain name record in question. Typically, such caching DNS servers also

implement the recursive algorithm necessary to resolve a given name starting with the DNS root through to the authoritative name servers of the queried domain. With this function implemented in the name server, user applications gain efficiency in design and operation.

The combination of DNS caching and recursive functions in a name server is not mandatory; the functions can be implemented independently in servers for special purposes.

Internet service providers typically provide recursive and caching name servers for their customers. In addition, many home networking routers implement DNS caches and recursors to improve efficiency in the local network.

DNS resolvers

The client side of the DNS is called a DNS resolver. A resolver is responsible for initiating and sequencing the queries that ultimately lead to a full resolution (translation) of the resource sought, e.g., translation of a domain name into an IP address. DNS resolvers are classified by a variety of query methods, such as *recursive*, *non-recursive*, and *iterative*. A resolution process may use a combination of these methods.[1]

In a *non-recursive query*, a DNS resolver queries a DNS server that provides a record either for which the server is authoritative, or it provides a partial result without querying other servers. In case of a caching DNS resolver, the non-recursive query of its local DNS cache delivers a result and reduces the load on upstream DNS servers by caching DNS resource records for a period of time after an initial response from upstream DNS servers.

In a *recursive query*, a DNS resolver queries a single DNS server, which may in turn query other DNS servers on behalf of the requester. For example, a simple stub resolver running on a home router typically makes a recursive query to the DNS server run by the user's ISP. A recursive query is one for which the DNS server answers the query completely by querying other name servers as needed. In typical operation, a client issues a recursive query to a caching recursive DNS server, which subsequently issues non-recursive queries to determine the answer and send a single answer back to the client. The resolver, or another DNS server acting recursively on behalf of the resolver, negotiates use of recursive service using bits in the query headers. DNS servers are not required to support recursive queries.

The *iterative query* procedure is a process in which a DNS resolver queries a chain of one or more DNS servers. Each server refers the client to the next server in the chain, until the current server can fully resolve the request. For example, a possible resolution of www.example.com would query a global root server, then a "com" server, and finally an "example.com" server.

Circular dependencies and glue records

Name servers in delegations are identified by name, rather than by IP address. This means that a resolving name server must issue another DNS request to find out the IP address of the server to which it has been referred. If the name given in the delegation is a subdomain of the domain for which the delegation is being provided, there is a circular dependency.

In this case, the name server providing the delegation must also provide one or more IP addresses for the authoritative name server mentioned in the delegation. This information is called *glue*. The delegating name server provides this glue in the form of records in the *additional section* of the DNS response, and provides the delegation in the *authority section* of the response. A glue record is a combination of the name server and IP address.

For example, if the authoritative name server for example.org is ns1.example.org, a computer trying to resolve www.example.org first resolves ns1.example.org. As ns1 is contained in example.org, this requires resolving example.org first, which presents a circular dependency. To break the dependency, the name server for the top level

domain org includes glue along with the delegation for example.org. The glue records are address records that provide IP addresses for ns1.example.org. The resolver uses one or more of these IP addresses to query one of the domain's authoritative servers, which allows it to complete the DNS query.

Record caching

A standard practice in implementing name resolution in applications is to reduce the load on the Domain Name System servers by caching results locally, or in intermediate resolver hosts. Results obtained from a DNS request are always associated with the time to live (TTL), an expiration time after which the results must be discarded or refreshed. The TTL is set by the administrator of the authoritative DNS server. The period of validity may vary from a few seconds to days or even weeks.

As a result of this distributed caching architecture, changes to DNS records do not propagate throughout the network immediately, but require all caches to expire and to be refreshed after the TTL. RFC 1912 conveys basic rules for determining appropriate TTL values.

Some resolvers may override TTL values, as the protocol supports caching for up to sixty-eight years or no caching at all. Negative caching, i.e. the caching of the fact of non-existence of a record, is determined by name servers authoritative for a zone which must include the Start of Authority (SOA) record when reporting no data of the requested type exists. The value of the *minimum* field of the SOA record and the TTL of the SOA itself is used to establish the TTL for the negative answer.

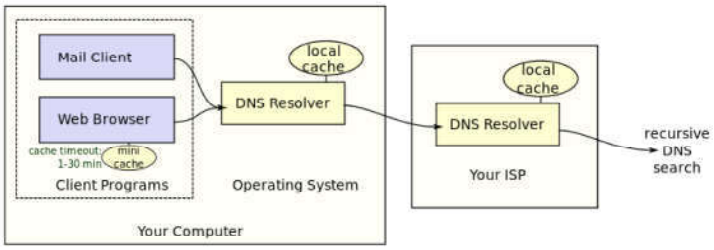
Reverse lookup

A reverse DNS lookup is a query of the DNS for domain names when the IP address is known. Multiple domain names may be associated with an IP address. The DNS stores IP addresses in the form of domain names as specially formatted names in pointer (PTR) records within the infrastructure top-level domain arpa. For IPv4, the domain is in-addr.arpa. For IPv6, the reverse lookup domain is ip6.arpa. The IP address is represented as a name in reverse-ordered octet representation for IPv4, and reverse-ordered nibble representation for IPv6.

When performing a reverse lookup, the DNS client converts the address into these formats before querying the name for a PTR record following the delegation chain as for any DNS query. For example, assuming the IPv4 address 208.80.152.2 is assigned to Wikimedia, it is represented as a DNS name in reverse order: 2.152.80.208.in-addr.arpa. When the DNS resolver gets a pointer (PTR) request, it begins by querying the root servers, which point to the servers of American Registry for Internet Numbers (ARIN) for the 208.in-addr.arpa zone. ARIN's servers delegate 152.80.208.in-addr.arpa to Wikimedia to which the resolver sends another query for 2.152.80.208.in-addr.arpa, which results in an authoritative response.

Client lookup

Users generally do not communicate directly with a DNS resolver. Instead DNS resolution takes place transparently in applications such as web browsers, e-mail clients, and other Internet applications. When an application makes a request that requires a domain name lookup, such programs send a resolution request to the DNS resolver in the local operating system, which in turn handles the communications required.



DNS resolution sequence

The DNS resolver will almost invariably have a cache (see above) containing recent lookups. If the cache can provide the answer to the request, the resolver will return the value in the cache to the program that made the request. If the cache does not contain the answer, the resolver will send the request to one or more designated DNS servers. In the case of most home users, the Internet service provider to which the machine connects will usually supply this DNS server: such a user will either have configured that server's address manually or allowed DHCP to set it; however, where systems administrators have configured systems to use their own DNS servers, their DNS resolvers point to separately maintained name servers of the organization. In any event, the name server thus queried will follow the process outlined above, until it either successfully finds a result or does not. It then returns its results to the DNS resolver; assuming it has found a result, the resolver duly caches that result for future use, and hands the result back to the software which initiated the request.

Broken resolvers

Some large ISPs have configured their DNS servers to violate rules, such as by disobeying TTLs, or by indicating that a domain name does not exist just because one of its name servers does not respond.[28]

Some applications, such as web browsers, maintain an internal DNS cache to avoid repeated lookups via the network. This practice can add extra difficulty when debugging DNS issues, as it obscures the history of such data. These caches typically use very short caching times – in the order of one minute.[29]

Internet Explorer represents a notable exception: versions up to IE 3.x cache DNS records for 24 hours by default. Internet Explorer 4.x and later versions (up to IE 8) decrease the default time out value to half an hour, which may be changed by modifying default configuration.[30]

Google Chrome triggers a specific error message for DNS issues. When the DNS server is down or broken, Google Chrome returns an error message.

Other applications

The Domain Name System includes several other functions and features.

Hostnames and IP addresses are not required to match in a one-to-one relationship. Multiple hostnames may correspond to a single IP address, which is useful in virtual hosting, in which many web sites are served from a single host. Alternatively, a single hostname may resolve to many IP addresses to facilitate fault tolerance and load distribution to multiple server instances across an enterprise or the global Internet.

DNS serves other purposes in addition to translating names to IP addresses. For instance, mail transfer agents use DNS to find the best mail server to deliver e-mail: An MX record provides a mapping between a domain and a mail exchanger; this can provide an additional layer of fault tolerance and load distribution.

The DNS is used for efficient storage and distribution of IP addresses of blacklisted email hosts. A common method is to place the IP address of the subject host into the sub-domain of a higher level domain name, and to resolve that name to a record that indicates a positive or a negative indication.

For example:

- The address 102.3.4.5 is blacklisted. It points to 5.4.3.102.blacklist.example, which resolves to 127.0.0.1.
- The address 102.3.4.6 is not blacklisted and points to 6.4.3.102.blacklist.example. This hostname is either not configured, or resolves to 127.0.0.2.

E-mail servers can query blacklist.example to find out if a specific host connecting to them is in the blacklist. Many of such blacklists, either subscription-based or free of cost, are available for use by email administrators and anti-spam software.

To provide resilience in the event of computer or network failure, multiple DNS servers are usually provided for coverage of each domain. At the top level of global DNS, thirteen groups of root name servers exist, with additional "copies" of them distributed worldwide via anycast addressing.

Dynamic DNS (DDNS) updates a DNS server with a client IP address on-the-fly, for example, when moving between ISPs or mobile hot spots, or when the IP address changes administratively.

DNS message format

The DNS protocol uses two types of DNS messages, queries and replies, and they both have the same format. Each message consists of a header and four sections: question, answer, authority, and an additional space. A header field (*flags*) controls the content of these four sections.[1]

The header section contains the following fields: *Identification*, *Flags*, *Number of questions*, *Number of answers*, *Number of authority resource records* (RRs), and *Number of additional RRs*. The identification field can be used to match responses with queries. The flag field consists of several sub-fields. The first is a single bit which indicates if the message is a query (0) or a reply (1). The second sub-field consists of four bits indicating the type of query, or the type of query this message is a response to. 0 is a standard query, 1 is an inverse query, 2 is a server status request. A single-bit sub-field indicates if the DNS server is authoritative for the queried hostname. Another single-bit sub-field indicates if the client wants to send a recursive query ("RD"). The next single-bit sub-field indicates if the replying DNS server supports recursion ("RA"), as not all DNS servers are configured to do this task. Another sub-field indicates if the message was truncated for some reason ("TC"), and a four-bit sub-field is used for error codes. The *question* section contains the domain name and type of record (A, AAAA, MX, TXT, etc.) being resolved. The domain name is broken into discrete labels which are concatenated; each label is prefixed by the length of that label. The *answer* section has the resource records of the queried name. A domain name may occur in multiple records if it has multiple IP addresses associated.[31]

DNS protocol transport

DNS primarily uses the User Datagram Protocol (UDP) on port number 53 to serve requests.[3] DNS queries consist of a single UDP request from the client followed by a single UDP reply from the server. When the length of the answer exceeds 512 bytes and both client and server support EDNS, larger UDP packets are used. Otherwise, the query is sent again using the Transmission Control Protocol (TCP). TCP is also used for tasks such as zone transfers. Some resolver implementations use TCP for all queries.

Resource records

The Domain Name System specifies a database of information elements for network resources. The types of information elements are categorized and organized with a list of DNS record types, the resource records (RRs). Each record has a type (name and number), an expiration time (time to live), a class, and type-specific data. Resource records of the same type are described as a *resource record set* (RRset), having no special ordering. DNS resolvers return the entire set upon query, but servers may implement round-robin ordering to achieve load balancing. In contrast, the Domain Name System Security Extensions (DNSSEC) work on the complete set of resource record in canonical order.

When sent over an Internet Protocol network, all records use the common format specified in RFC 1035:[32]

Resource record (RR) fields

Field	Description	Length (octets)
NAME	Name of the node to which this record pertains	Variable
TYPE	Type of RR in numeric form (e.g., 15 for MX RRs)	2
CLASS	Class code	2
TTL	Count of seconds that the RR stays valid (The maximum is $2^{31}-1$, which is about 68 years)	4
RDLENGTH	Length of RDATA field (specified in octets)	2
RDATA	Additional RR-specific data	Variable, as per RDLENGTH

NAME is the fully qualified domain name of the node in the tree. On the wire, the name may be shortened using label compression where ends of domain names mentioned earlier in the packet can be substituted for the end of the current domain name. A free standing @ is used to denote the current origin.

TYPE is the record type. It indicates the format of the data and it gives a hint of its intended use. For example, the *A* record is used to translate from a domain name to an IPv4 address, the *NS* record lists which name servers can answer lookups on a DNS zone, and the *MX* record specifies the mail server used to handle mail for a domain specified in an e-mail address.

RDATA is data of type-specific relevance, such as the IP address for address records, or the priority and hostname for MX records. Well known record types may use label compression in the RDATA field, but "unknown" record types must not (RFC 3597).

The *CLASS* of a record is set to IN (for *Internet*) for common DNS records involving Internet hostnames, servers, or IP addresses. In addition, the classes Chaos (CH) and Hesiod (HS) exist.[33] Each class is an independent name space with potentially different delegations of DNS zones.

In addition to resource records defined in a zone file, the domain name system also defines several request types that are used only in communication with other DNS nodes (*on the wire*), such as when performing zone transfers (AXFR/IXFR) or for EDNS (OPT).

Wildcard DNS records

The domain name system supports wildcard DNS records which specify names that start with the *asterisk label*, '*', e.g., *.example.[1][34] DNS records belonging to wildcard domain names specify rules for generating resource records within a single DNS zone by substituting whole labels with matching components of the query name, including any specified descendants. For example, in the following configuration, the DNS zone *x.example* specifies that all subdomains, including subdomains of subdomains, of *x.example* use the mail exchanger (MX) *a.x.example*. The A record for *a.x.example* is needed to specify the mail exchanger IP address. As this has the result of excluding this domain name and its subdomains from the wildcard matches, an additional MX record for the subdomain *a.x.example*, as well as a wildcarded MX record for all of its subdomains, must also be defined in the DNS zone.

```
x.example.        MX   10 a.x.example.
*.x.example.      MX   10 a.x.example.
*.a.x.example.    MX   10 a.x.example.
a.x.example.      MX   10 a.x.example.
a.x.example.      AAAA 2001:db8::1
```

The role of wildcard records was refined in RFC 4592, because the original definition in RFC 1034 was incomplete and resulted in misinterpretations by implementers.[34]

Protocol extensions

The original DNS protocol had limited provisions for extension with new features. In 1999, Paul Vixie published in RFC 2671 (superseded by RFC 6891) an extension mechanism, called Extension mechanisms for DNS (EDNS) that introduced optional protocol elements without increasing overhead when not in use. This was accomplished through the OPT pseudo-resource record that only exists in wire transmissions of the protocol, but not in any zone files. Initial extensions were also suggested (EDNS0), such as increasing the DNS message size in UDP datagrams.

Dynamic zone updates

Dynamic DNS updates use the UPDATE DNS opcode to add or remove resource records dynamically from a zone database maintained on an authoritative DNS server. The feature is described in RFC 2136. This facility is useful to register network clients into the DNS when they boot or become otherwise available on the network. As a booting client may be assigned a different IP address each time from a DHCP server, it is not possible to provide static DNS assignments for such clients.

Security issues

Originally, security concerns were not major design considerations for DNS software or any software for deployment on the early Internet, as the network was not open for participation by the general public. However, the expansion of the Internet into the commercial sector in the 1990s changed the requirements for security measures to protect data integrity and user authentication.

Several vulnerability issues were discovered and exploited by malicious users. One such issue is DNS cache poisoning, in which data is distributed to caching resolvers under the pretense of being an authoritative origin server, thereby polluting the data store with potentially false information and long expiration times (time-to-live). Subsequently, legitimate application requests may be redirected to network hosts operated with malicious intent.

DNS responses traditionally do not have a cryptographic signature, leading to many attack possibilities; the Domain Name System Security Extensions (DNSSEC) modify DNS to add support for cryptographically signed responses. DNSCurve has been proposed as an alternative to DNSSEC. Other extensions, such as TSIG, add support for cryptographic authentication between trusted peers and are commonly used to authorize zone transfer or dynamic update operations.

Some domain names may be used to achieve spoofing effects. For example, paypal.com and paypa1.com are different names, yet users may be unable to distinguish them in a graphical user interface depending on the user's chosen typeface. In many fonts the letter *l* and the numeral *1* look very similar or even identical. This problem is acute in systems that support internationalized domain names, as many character codes in ISO 10646 may appear identical on typical computer screens. This vulnerability is occasionally exploited in phishing.[35]

Techniques such as forward-confirmed reverse DNS can also be used to help validate DNS results.

Privacy and tracking issues

A device looking up a DNS record must communicate with a DNS server to do so. Considerable attention has been given to the adverse privacy implications. Even if DNS records cannot easily be read, modified or spoofed due to security extensions, a person with access to the DNS server or the traffic stream "on the wire" may have little difficulty

in matching the IP address of the device (which often identifies the user), to the websites, email or other domains they visit, and track how often and when these records are queried, since DNS records typically expire and must be requeried regularly.

DNS can also "leak" from otherwise secure or private connections, if attention is not paid to their configuration, and at times DNS has been used to bypass firewalls by malicious persons, and exfiltrate data, since it is often seen as innocuous.

Two main approaches are in use to counter privacy issues with DNS:

- Proxies (including Tor) and VPNs which can reroute or anonymize DNS inquiries in order to mask the source IP address.
- Intermediate DNS servers that are configured with minimal logging, and which can be queried instead of querying usual DNS servers whose privacy policy may be untrusted. In 2018, Cloudflare launched a DNS server of this kind.[36]

Domain name registration

The right to use a domain name is delegated by domain name registrars which are accredited by the Internet Corporation for Assigned Names and Numbers (ICANN) or other organizations such as OpenNIC, that are charged with overseeing the name and number systems of the Internet. In addition to ICANN, each top-level domain (TLD) is maintained and serviced technically by an administrative organization, operating a registry. A *registry* is responsible for operating the database of names within its authoritative zone, although the term is most often used for TLDs. A *registrant* is a person or organization who asked for domain registration.[22] The registry receives registration information from each domain name *registrar*, which is authorized (accredited) to assign names in the corresponding zone and publishes the information using the WHOIS protocol. As of 2015, usage of RDAP is being considered.[37]

ICANN publishes the complete list of TLDs, TLD registries, and domain name registrars. Registrant information associated with domain names is maintained in an online database accessible with the WHOIS service. For most of the more than 290 country code top-level domains (ccTLDs), the domain registries maintain the WHOIS (Registrant, name servers, expiration dates, etc.) information. For instance, DENIC, Germany NIC, holds the DE domain data. From about 2001, most Generic top-level domain (gTLD) registries have adopted this so-called *thick* registry approach, i.e. keeping the WHOIS data in central registries instead of registrar databases.

For top-level domains on COM and NET, a *thin* registry model is used. The domain registry (e.g., GoDaddy, BigRock and PDR, VeriSign, etc., etc.) holds basic WHOIS data (i.e., registrar and name servers, etc.). Organizations, or registrants using ORG on the other hand, are on the Public Interest Registry exclusively.

Some domain name registries, often called *network information centers* (NIC), also function as registrars to end-users, in addition to providing access to the WHOIS datasets. The top-level domain registries, such as for the domains COM, NET, and ORG use a registry-registrar model consisting of many domain name registrars.[38] In this method of management, the registry only manages the domain name database and the relationship with the registrars. The *registrants* (users of a domain name) are customers of the registrar, in some cases through additional subcontracting of resellers.

RFC documents

Standards

The Domain Name System is defined by Request for Comments (RFC) documents published by the Internet Engineering Task Force (Internet standards). The following is a list of RFCs that define the DNS protocol.

- RFC 1034, *Domain Names - Concepts and Facilities*
- RFC 1035, *Domain Names - Implementation and Specification*
- RFC 1123, *Requirements for Internet Hosts—Application and Support*
- RFC 1995, *Incremental Zone Transfer in DNS*
- RFC 1996, *A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)*
- RFC 2136, *Dynamic Updates in the domain name system (DNS UPDATE)*
- RFC 2181, *Clarifications to the DNS Specification*
- RFC 2308, *Negative Caching of DNS Queries (DNS NCACHE)*
- RFC 2672, *Non-Terminal DNS Name Redirection*
- RFC 2845, *Secret Key Transaction Authentication for DNS (TSIG)*
- RFC 3225, *Indicating Resolver Support of DNSSEC*
- RFC 3226, *DNSSEC and IPv6 A6 aware server/resolver message size requirements*
- RFC 3596, *DNS Extensions to Support IP Version 6*
- RFC 3597, *Handling of Unknown DNS Resource Record (RR) Types*
- RFC 4343, *Domain Name System (DNS) Case Insensitivity Clarification*
- RFC 4592, *The Role of Wildcards in the Domain Name System*
- RFC 4635, *HMAC SHA TSIG Algorithm Identifiers*
- RFC 5001, *DNS Name Server Identifier (NSID) Option*
- RFC 5011, *Automated Updates of DNS Security (DNSSEC) Trust Anchors*
- RFC 5452, *Measures for Making DNS More Resilient against Forged Answers*
- RFC 5890, *Internationalized Domain Names for Applications (IDNA):Definitions and Document Framework*
- RFC 5891, *Internationalized Domain Names in Applications (IDNA): Protocol*
- RFC 5892, *The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)*
- RFC 5893, *Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)*
- RFC 6891, *Extension Mechanisms for DNS (EDNS0)*
- RFC 7766, *DNS Transport over TCP - Implementation Requirements*

Proposed security standards

- RFC 4033, *DNS Security Introduction and Requirements*
- RFC 4034, *Resource Records for the DNS Security Extensions*
- RFC 4035, *Protocol Modifications for the DNS Security Extensions*
- RFC 4509, *Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records*
- RFC 4470, *Minimally Covering NSEC Records and DNSSEC On-line Signing*
- RFC 5155, *DNS Security (DNSSEC) Hashed Authenticated Denial of Existence*
- RFC 5702, *Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC*
- RFC 5910, *Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)*
- RFC 5933, *Use of GOST Signature Algorithms in DNSKEY and RRSIG Resource Records for DNSSEC*
- RFC 7830, *The EDNS(0) Padding Option*
- RFC 7858, *Specification for DNS over Transport Layer Security (TLS)*
- RFC 8310, *Usage Profiles for DNS over TLS and DNS over DTLS*
- RFC 8484, *DNS Queries over HTTPS (DoH)*

Experimental RFCs

- RFC 1183, *New DNS RR Definitions*

Best Current Practices

- RFC 2182, *Selection and Operation of Secondary DNS Servers* (BCP 16)
- RFC 2317, *Classless IN-ADDR.ARPA delegation* (BCP 20)
- RFC 5625, *DNS Proxy Implementation Guidelines* (BCP 152)
- RFC 6895, *Domain Name System (DNS) IANA Considerations* (BCP 42)
- RFC 7720, *DNS Root Name Service Protocol and Deployment Requirements* (BCP 40)

Informational RFCs

These RFCs are advisory in nature, but may provide useful information despite defining neither a standard or BCP. (RFC 1796)

- RFC 1178, *Choosing a Name for Your Computer* (FYI 5)
- RFC 1591, *Domain Name System Structure and Delegation*
- RFC 1912, *Common DNS Operational and Configuration Errors*
- RFC 2100, *The Naming of Hosts*
- RFC 3696, *Application Techniques for Checking and Transformation of Names*
- RFC 4892, *Requirements for a Mechanism Identifying a Name Server Instance*
- RFC 5894, *Internationalized Domain Names for Applications (IDNA):Background, Explanation, and Rationale*
- RFC 5895, *Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008*
- RFC 7626, *DNS Privacy Considerations*
- RFC 7706, *Decreasing Access Time to Root Servers by Running One on Loopback*

Unknown

These RFCs have an official status of Unknown, but due to their age are not clearly labeled as such.

- RFC 920, *Domain Requirements* – Specified original top-level domains
- RFC 1032, *Domain Administrators Guide*
- RFC 1033, *Domain Administrators Operations Guide*
- RFC 1101, *DNS Encodings of Network Names and Other Types*

See also

- Alternative DNS root
- Comparison of DNS server software
- Domain hijacking
- DNS hijacking
- DNS management software
- DNS over HTTPS
- DNS over TLS
- Hierarchical namespace
- IPv6 brokenness and DNS whitelisting
- Multicast DNS
- Public recursive name server
- resolv.conf
- Split-horizon DNS
- List of DNS record types
- List of managed DNS providers

References

1. RFC 1034, *Domain Names - Concepts and Facilities*, P. Mockapetris, The Internet Society (November 1987)
2. RFC 781, *Internet Protocol - DARPA Internet Program Protocol Specification*, Information Sciences Institute, J. Postel (Ed.), The Internet Society (September 1981)

3. RFC 1035, *Domain Names - Implementation and Specification*, P. Mockapetris, The Internet Society (November 1987)
4. J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl. "Globally Distributed Content Delivery, IEEE Internet Computing, September/October 2002, pp. 50-58" (https://people.cs.umass.edu/~ramesh/Site/PUBLICATIONS_files/DMPPSW02.pdf) (PDF).
5. Nygren., E.; Sitaraman R. K.; Sun, J. (2010). "The Akamai Network: A Platform for High-Performance Internet Applications" (http://www.akamai.com/dl/technical_publications/network_overview_osr.pdf) (PDF). *ACM SIGOPS Operating Systems Review*. **44** (3): 2–19. doi:10.1145/1842733.1842736 (https://doi.org/10.1145%2F1842733.1842736). Retrieved November 19, 2012.
6. Paul Mockapetris (November 1987). "SOA RDATA format" (https://tools.ietf.org/html/rfc1035#section-3.3.13). *Domain Names - Implementation and Specification* (https://tools.ietf.org/html/rfc1035). IETF. sec. 3.3.13. doi:10.17487/RFC1035 (https://doi.org/10.17487%2FRFC1035). RFC 1035. Retrieved 18 December 2015.
7. Champika Wijayatunga (February 2015). "DNS Abuse Handling" (https://conference.apnic.net/data/39/dns-abuse-handling-final_1425362607.pdf) (PDF). APNIC. Retrieved 18 December 2016.
8. RFC 3467, "Role of the Domain Name System (DNS)", J.C. Klensin, J. Klensin (February 2003).
9. Liu, Cricket; Albitz, Paul (2006). *DNS and BIND* (5th ed.). O'Reilly Media. p. 3. ISBN 978-0-596-10057-5.
10. Evans 2018, p. 112.
11. Evans 2018, p. 113.
12. IEEE Annals [3B2-9] man2011030074.3d 29/7/011 11:54 Page 74
13. "Why Does the Net Still Work on Christmas? Paul Mockapetris - Internet Hall of Fame" (http://internethalloffame.org/blog/2012/07/23/why-does-net-still-work-christmas-paul-mockapetris). *internethalloffame.org*.
14. Evans 2018, p. 119.
15. Evans 2018, p. 120.
16. Evans 2018, p. 120-121.
17. "Elizabeth Feinler" (https://web.archive.org/web/20180914182353/https://www.internethalloffame.org/inductees/elizabeth-feinler). *Internet Hall of Fame*. Archived from the original (https://www.internethalloffame.org/inductees/elizabeth-feinler) on 14 September 2018. Retrieved 2018-11-25.
18. Andrei Robachevsky (26 November 2013). "Happy 30th Birthday, DNS!" (http://www.internetsociety.org/blog/2013/11/happy-30th-birthday-dns). Internet Society. Retrieved 18 December 2015.
19. Elizabeth Feinler, IEEE Annals, 3B2-9 man2011030074.3d 29/7/011 11:54 Page 74
20. Terry, Douglas B.; et al. (June 12–15, 1984). "The Berkeley Internet Name Domain Server" (http://www.eecs.berkeley.edu/Pubs/TechRpts/1984/5957.html). *Summer Conference, Salt Lake City 1984: Proceedings*. USENIX Association Software Tools Users Group. pp. 23–31.
21. Internet Systems Consortium. "The Most Widely Used Name Server Software: BIND" (https://www.isc.org/downloads/bind/). History of BIND. Retrieved 28 July 2013.
22. Paul Hoffman; Andrew Sullivan; Kazunori Fujiwara (December 2015). *DNS Terminology* (https://tools.ietf.org/html/rfc7719). IETF. doi:10.17487/RFC7719 (https://doi.org/10.17487%2FRFC7719). RFC 7719. Retrieved 18 December 2015.
23. Paul Mockapetris (November 1987). "Name space specifications and terminology" (https://tools.ietf.org/html/rfc1034#section-3.1). *Domain Names - Domain Concepts and Facilities* (https://tools.ietf.org/html/rfc1034). IETF. sec. 3.1. doi:10.17487/RFC1034 (https://doi.org/10.17487%2FRFC1034). RFC 1034. Retrieved 17 December 2015.
24. Paul Mockapetris (November 1987). "How the database is divided into zones" (https://tools.ietf.org/html/rfc1034#section-4.2). *Domain Names - Domain Concepts and Facilities* (https://tools.ietf.org/html/rfc1034). IETF. sec. 4.2. doi:10.17487/RFC1034 (https://doi.org/10.17487%2FRFC1034). RFC 1034. Retrieved 17 December 2015.
25. Lindsay, David (2007). *International Domain Name Law: ICANN and the UDRP*. Bloomsbury Publishing. p. 8. ISBN 978-1-84113-584-7.
26. Network Working Group of the IETF, January 2006, RFC 4343: Domain Name System (DNS) Case Insensitivity Clarification

27. RFC 3696, *Application Techniques for Checking and Transformation of Names*, J. Klensin
28. "Providers ignoring DNS TTL?" (http://ask.slashdot.org/story/05/04/18/198259/providers-ignoring-dns-ttl). Slashdot. 2005. Retrieved 2012-04-07.
29. Ben Anderson (7 September 2011). "Ben Anderson: Why Web Browser DNS Caching Can Be A Bad Thing" (http://dyn.com/web-browser-dns-caching-bad-thing/). Retrieved 20 October 2014.
30. "How Internet Explorer uses the cache for DNS host entries" (http://support.microsoft.com/default.aspx?scid=KB;en-us;263558). Microsoft Corporation. 2004. Retrieved 2010-07-25.
31. James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach, 6th ed. Essex, England: Pearson Educ. Limited, 2012
32. RFC 5395, *Domain Name System (DNS) IANA Considerations*, D. Eastlake 3rd (November 2008), Section 3
33. RFC 5395, *Domain Name System (DNS) IANA Considerations*, D. Eastlake 3rd (November 2008), p. 11
34. RFC 4592, *The Role of Wildcards in the Domain Name System*, E. Lewis (July 2006)
35. APWG. "Global Phishing Survey: Domain Name Use and Trends in 1H2010." 10/15/2010 apwg.org (http://www.apwg.org/reports/APWG_GlobalPhishingSurvey_1H2010.pdf)
36. *Cloudflare Launches 1.1.1.1 DNS Service to Improve Internet Privacy* (http://www.eweek.com/security/cloudflare-launches-1.1.1.1-dns-service-to-improve-internet-privacy), retrieved 2018-04-20
37. "Registration Data Access Protocol (RDAP) Operational Profile for gTLD Registries and Registrars" (https://web.archive.org/web/20151222144443/https://www.icann.org/news/announcement-2015-12-03-en). ICANN. 3 December 2015. Archived from the original (https://www.icann.org/news/announcement-2015-12-03-en) on 22 December 2015. Retrieved 18 December 2015.
38. "Find a Registrar" (http://www.verisign.com/en_US/domain-names/domain-registrar/index.xhtml). VeriSign, Inc. Retrieved 18 December 2015.

Sources

- Evans, Claire L. (2018). *Broad Band: The Untold Story of the Women Who Made the Internet* (https://books.google.com/books?id=C8ouDwAAQBAJ&lpg=PP1&dq=9780735211759&pg=PP1#v=onepage&q=9780735211759&f=false). New York: Portfolio/Penguin. ISBN 9780735211759.

External links

- Vixie, Paul (2007-04-01). "DNS Complexity" (https://web.archive.org/web/20070610092333/http://acmqueue.com/modules.php?name=Content&pa=showpage&pid=481). ACM Queue. Archived from the original (http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=481) on 2007-06-10.
- Zytrax.com (http://www.zytrax.com/books/dns/), Open Source Guide – DNS for Rocket Scientists.
- Internet Governance and the Domain Name System: Issues for Congress (http://www.fas.org/sgp/crs/misc/R42351.pdf) Congressional Research Service
- Ball, James (28 February 2014). "Meet the seven people who hold the keys to worldwide internet security" (https://www.theguardian.com/technology/2014/feb/28/seven-people-keys-worldwide-internet-security-web). *The Guardian*. Guardian News & Media Limited. Retrieved 28 February 2014.

WIKIPEDIA

Secure Shell

Secure Shell (**SSH**) is a cryptographic network protocol for operating network services securely over an unsecured network.[1] Typical applications include remote command-line login and remote command execution, but any network service can be secured with SSH.

SSH provides a secure channel over an unsecured network in a client–server architecture, connecting an SSH client application with an SSH server.[2] The protocol specification distinguishes between two major versions, referred to as SSH-1 and SSH-2. The standard TCP port for SSH is 22. SSH is generally used to access Unix-like operating systems, but it can also be used on Microsoft Windows. Windows 10 uses OpenSSH as its default SSH client.[3]

SSH was designed as a replacement for Telnet and for unsecured remote shell protocols such as the Berkeley rlogin, rsh, and rexec protocols. Those protocols send information, notably passwords, in plaintext, rendering them susceptible to interception and disclosure using packet analysis.[4] The encryption used by SSH is intended to provide confidentiality and integrity of data over an unsecured network, such as the Internet, although files leaked by Edward Snowden indicate that the National Security Agency can sometimes decrypt SSH, allowing them to read the contents of SSH sessions.[5]

Contents

Definition

SSH uses public-key cryptography to authenticate the remote computer and allow it to authenticate the user, if necessary.[2] There are several ways to use SSH; one is to use automatically generated public-private key pairs to simply encrypt a network connection, and then use password authentication to log on.

Another is to use a manually generated public-private key pair to perform the authentication, allowing users or programs to log in without having to specify a password. In this scenario, anyone can produce a matching pair of different keys (public and private). The public key is placed on all computers that must allow access to the owner of the matching private key (the owner keeps the private key secret). While authentication is based on the private key, the key itself is never transferred through the network during authentication. SSH only verifies whether the same person offering the public key also owns the matching private key. In all versions of SSH it is important to verify unknown public keys, i.e. associate the public keys with identities, before accepting them as valid. Accepting an attacker's public key without validation will authorize an unauthorized attacker as a valid user.

Key management

On Unix-like systems, the list of authorized public keys is typically stored in the home directory of the user that is allowed to log in remotely, in the file ~/.ssh/authorized_keys.[6] This file is respected by SSH only if it is not writable by anything apart from the owner and root. When the public key is present on the remote end and the matching private key is present on the local end, typing in the password is no longer required (some software like Message Passing Interface (MPI) stack may need this password-less access to run properly). However, for additional security the private key itself can be locked with a passphrase.

The private key can also be looked for in standard places, and its full path can be specified as a command line setting (the option *-i* for ssh). The ssh-keygen utility produces the public and private keys, always in pairs.

SSH also supports password-based authentication that is encrypted by automatically generated keys. In this case, the attacker could imitate the legitimate server side, ask for the password, and obtain it (man-in-the-middle attack). However, this is possible only if the two sides have never authenticated before, as SSH remembers the key that the server side previously used. The SSH client raises a warning before accepting the key of a new, previously unknown server. Password authentication can be disabled.

Usage

SSH is typically used to log into a remote machine and execute commands, but it also supports tunneling, forwarding TCP ports and X11 connections; it can transfer files using the associated SSH file transfer (SFTP) or secure copy (SCP) protocols.[2] SSH uses the client-server model.

The standard TCP port 22 has been assigned for contacting SSH servers.[7]

An SSH client program is typically used for establishing connections to an SSH daemon accepting remote connections. Both are commonly present on most modern operating systems, including macOS, most distributions of Linux, OpenBSD, FreeBSD, NetBSD, Solaris and OpenVMS. Notably, versions of Windows prior to 1709 do not include SSH by default. Proprietary, freeware and open source (e.g. PuTTY,[8] and the version of OpenSSH which is part of Cygwin[9]) versions of various levels of complexity and completeness exist. File managers for UNIX-like systems (e.g. Konqueror) can use the FISH protocol to provide a split-pane GUI with drag-and-drop. The open source Windows program WinSCP[10] provides similar file management (synchronization, copy, remote delete) capability using PuTTY as a back-end. Both WinSCP[11] and PuTTY[12] are available packaged to run directly off a USB drive, without requiring installation on the client machine. Setting up an SSH server in Windows typically involves enabling a feature in Settings app. In Windows 10 version 1709, an official Win32 port of OpenSSH is available.

SSH is important in cloud computing to solve connectivity problems, avoiding the security issues of exposing a cloud-based virtual machine directly on the Internet. An SSH tunnel can provide a secure path over the Internet, through a firewall to a virtual machine.[13]

History and development

Version 1.x

In 1995, Tatu Ylönen, a researcher at Helsinki University of Technology, Finland, designed the first version of the protocol (now called **SSH-1**) prompted by a password-sniffing attack at his university network.[14] The goal of SSH was to replace the earlier rlogin, TELNET, FTP[15] and rsh protocols, which did not provide strong authentication nor guarantee confidentiality. Ylönen released his implementation as freeware in July 1995, and the tool quickly gained in popularity. Towards the end of 1995, the SSH user base had grown to 20,000 users in fifty countries.

In December 1995, Ylönen founded SSH Communications Security to market and develop SSH. The original version of the SSH software used various pieces of free software, such as GNU libgmp, but later versions released by SSH Communications Security evolved into increasingly proprietary software.

It was estimated that by the year 2000 the number of users had grown to 2 million.[16]

Version 2.x

"Secsh" was the official Internet Engineering Task Force's (IETF) name for the IETF working group responsible for version 2 of the SSH protocol.[17] In 2006, a revised version of the protocol, **SSH-2**, was adopted as a standard. This version is incompatible with SSH-1. SSH-2 features both security and feature improvements over SSH-1. Better security, for example, comes through Diffie–Hellman key exchange and strong integrity checking via message authentication codes. New features of SSH-2 include the ability to run any number of shell sessions over a single SSH connection.[18] Due to SSH-2's superiority and popularity over SSH-1, some implementations such as libssh(v0.8.0+)[19], Lsh[20] and Dropbear[21] support only the SSH-2 protocol.

Version 1.99

In January 2006, well after version 2.1 was established, RFC 4253 specified that an SSH server which supports both 2.0 and prior versions of SSH should identify its protoversion as 1.99.[22] This is not an actual version but a method to identify backward compatibility.

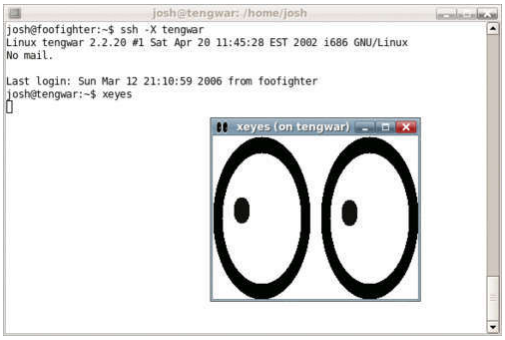
OpenSSH and OSSH

In 1999, developers, wanting a free software version to be available, went back to the older 1.2.12 release of the original SSH program, which was the last released under an open source license. Björn Grönvall's OSSH was subsequently developed from this codebase. Shortly thereafter, OpenBSD developers forked Grönvall's code and did extensive work on it, creating OpenSSH, which shipped with the 2.6 release of OpenBSD. From this version, a "portability" branch was formed to port OpenSSH to other operating systems.[23]

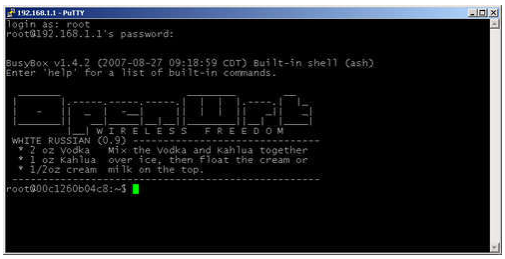
As of 2005, OpenSSH was the single most popular SSH implementation, coming by default in a large number of operating systems. OSSH meanwhile has become obsolete.[24] OpenSSH continues to be maintained and supports the SSH-2 protocol, having expunged SSH-1 support from the codebase with the OpenSSH 7.6 release.

Uses

SSH is a protocol that can be used for many applications across many platforms including most Unix variants (Linux, the BSDs including Apple's macOS, and Solaris), as well as Microsoft Windows. Some of the applications below may require features that are only available or compatible with specific SSH clients or servers. For example, using the SSH protocol to implement a VPN is possible, but presently only with the OpenSSH server and client implementation.



Example of tunneling an X11 application over SSH: the user 'josh' has SSHed from the local machine 'foofighter' to the remote machine 'tengwar' to run xeyes.



Logging into OpenWrt via SSH using PuTTY running on Windows.

- For login to a shell on a remote host (replacing Telnet and rlogin)
- For executing a single command on a remote host (replacing rsh)
- For setting up automatic (passwordless) login to a remote server (for example, using OpenSSH[25])
- In combination with rsync to back up, copy and mirror files efficiently and securely
- For forwarding or tunneling a port (not to be confused with a VPN, which routes packets between different networks, or bridges two broadcast domains into one).
- For using as a full-fledged encrypted VPN. Note that only OpenSSH server and client supports this feature.
- For forwarding X from a remote host (possible through multiple intermediate hosts)
- For browsing the web through an encrypted proxy connection with SSH clients that support the SOCKS protocol.
- For securely mounting a directory on a remote server as a filesystem on a local computer using SSHFS.
- For automated remote monitoring and management of servers through one or more of the mechanisms discussed above.
- For development on a mobile or embedded device that supports SSH.
- For securing file transfer protocols.

File transfer protocols

The Secure Shell protocols are used in several file transfer mechanisms.

- Secure copy (SCP), which evolved from RCP protocol over SSH
- rsync, intended to be more efficient than SCP. Generally runs over an SSH connection.
- SSH File Transfer Protocol (SFTP), a secure alternative to FTP (not to be confused with FTP over SSH or FTPS)
- Files transferred over shell protocol (a.k.a. FISH), released in 1998, which evolved from Unix shell commands over SSH
- Fast and Secure Protocol (FASP), aka *Aspera*, uses SSH for control and UDP ports for data transfer.

Architecture

The SSH-2 protocol has an internal architecture (defined in RFC 4251) with well-separated layers, namely:

- The *transport* layer (RFC 4253), which typically runs on top of TCP/IP. This layer handles initial key exchange as well as server authentication, and sets up encryption, compression and integrity verification. It exposes to the upper layer an interface for sending and receiving plaintext packets with sizes of up to 32,768 bytes each (more can be allowed by the implementation). The transport layer also arranges for key re-exchange, usually after 1 GB of data has been transferred or after 1 hour has passed, whichever occurs first.
- The *user authentication* layer (RFC 4252). This layer handles client authentication and provides a number of authentication methods. Authentication is *client-driven*: when one is prompted for a password, it may be the SSH client prompting, not the server. The server merely responds to the client's authentication requests. Widely used user-authentication methods include the following:
  - *password*: a method for straightforward password authentication, including a facility allowing a password to be changed. Not all programs implement this method.
  - *publickey*: a method for public key-based authentication, usually supporting at least DSA, ECDSA or RSA keypairs, with other implementations also supporting X.509 certificates.
  - *keyboard-interactive* (RFC 4256): a versatile method where the server sends one or more prompts to enter information and the client displays them and sends back responses keyed-in by the user. Used to provide one-time password authentication such as S/Key or SecurID. Used by some OpenSSH configurations when PAM is the underlying host-authentication provider to effectively provide password authentication, sometimes leading to inability to log in with a client that supports just the plain *password* authentication method.
  - GSSAPI authentication methods which provide an extensible scheme to perform SSH authentication using external mechanisms such as Kerberos 5 or NTLM, providing single sign-on capability to SSH sessions. These methods are usually implemented by commercial SSH implementations for use in organizations, though OpenSSH does have a working GSSAPI implementation.
- The *connection* layer (RFC 4254). This layer defines the concept of channels, channel requests and global requests using which SSH services are provided. A single SSH connection can host multiple channels simultaneously, each transferring data in both directions. Channel requests are used to relay out-of-band channel-specific data, such as the changed size of a terminal window or the exit code of a server-side process. Additionally, each channel performs its own flow control using the receive window size. The SSH client requests a server-side port to be forwarded using a global request. Standard channel types include:
  - *shell* for terminal shells, SFTP and exec requests (including SCP transfers)
  - *direct-tcpip* for client-to-server forwarded connections
  - *forwarded-tcpip* for server-to-client forwarded connections
- The SSHFP DNS record (RFC 4255) provides the public host key fingerprints in order to aid in verifying the authenticity of the host.

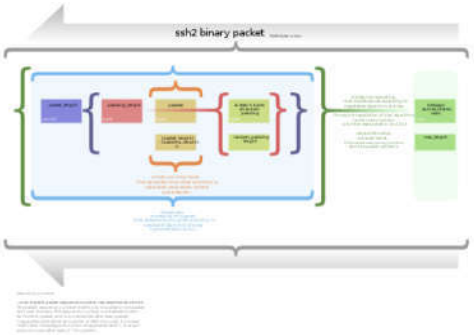


Diagram of the SSH-2 binary packet.

This open architecture provides considerable flexibility, allowing the use of SSH for a variety of purposes beyond a secure shell. The functionality of the transport layer alone is comparable to Transport Layer Security (TLS); the user-authentication layer is highly extensible with custom authentication methods; and the connection layer provides the ability to multiplex many secondary sessions into a single SSH connection, a feature comparable to BEEP and not available in TLS.

Enhancements

These are intended for performance enhancements of SSH products:

- SSH-over-SCTP: support for SCTP rather than TCP as the connection oriented transport layer protocol.[26]
- ECDSA: support for elliptic curve DSA rather than DSA or RSA for signing.[27]

- ECDH: support for elliptic curve Diffie–Hellman rather than plain Diffie–Hellman for encryption key exchange.[27]
- UMAC: support for UMAC rather than HMAC for MAC/integrity.[28]

Vulnerabilities

SSH-1

In 1998 a vulnerability was described in SSH 1.5 which allowed the unauthorized insertion of content into an encrypted SSH stream due to insufficient data integrity protection from CRC-32 used in this version of the protocol.[29][30] A fix known as SSH Compensation Attack Detector[31] was introduced into most implementations. Many of these updated implementations contained a new integer overflow vulnerability[32] that allowed attackers to execute arbitrary code with the privileges of the SSH daemon, typically root.

In January 2001 a vulnerability was discovered that allows attackers to modify the last block of an IDEA-encrypted session.[33] The same month, another vulnerability was discovered that allowed a malicious server to forward a client authentication to another server.[34]

Since SSH-1 has inherent design flaws which make it vulnerable, it is now generally considered obsolete and should be avoided by explicitly disabling fallback to SSH-1. Most modern servers and clients support SSH-2.

CBC plaintext recovery

In November 2008, a theoretical vulnerability was discovered for all versions of SSH which allowed recovery of up to 32 bits of plaintext from a block of ciphertext that was encrypted using what was then the standard default encryption mode, CBC.[35] The most straightforward solution is to use CTR, counter mode, instead of CBC mode, since this renders SSH resistant to the attack.[35]

Possible vulnerabilities

On December 28, 2014 Der Spiegel published classified information[5] leaked by whistleblower Edward Snowden which suggests that the National Security Agency may be able to decrypt some SSH traffic. The technical details associated with such a process were not disclosed.

An analysis in 2017 of the hacking tools BothanSpy & Gyrfalcon suggested that the SSH protocol itself was not compromised.[36]

Standards documentation

The following RFC publications by the IETF "secsh" working group document SSH-2 as a proposed Internet standard.

- RFC 4250, The Secure Shell (SSH) Protocol Assigned Numbers
- RFC 4251, The Secure Shell (SSH) Protocol Architecture
- RFC 4252, The Secure Shell (SSH) Authentication Protocol
- RFC 4253, The Secure Shell (SSH) Transport Layer Protocol
- RFC 4254, The Secure Shell (SSH) Connection Protocol
- RFC 4255, Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints
- RFC 4256, Generic Message Exchange Authentication for the Secure Shell Protocol (SSH)
- RFC 4335, The Secure Shell (SSH) Session Channel Break Extension
- RFC 4344, The Secure Shell (SSH) Transport Layer Encryption Modes
- RFC 4345, Improved Arcfour Modes for the Secure Shell (SSH) Transport Layer Protocol

It was later modified and expanded by the following publications.

- RFC 4419, Diffie-Hellman Group Exchange for the Secure Shell (SSH) Transport Layer Protocol (March 2006)
- RFC 4432, RSA Key Exchange for the Secure Shell (SSH) Transport Layer Protocol (March 2006)
- RFC 4462, Generic Security Service Application Program Interface (GSS-API) Authentication and Key Exchange for the Secure Shell (SSH) Protocol (May 2006)
- RFC 4716, The Secure Shell (SSH) Public Key File Format (November 2006)
- RFC 4819: Secure Shell Public Key Subsystem (March 2007)
- RFC 5647: AES Galois Counter Mode for the Secure Shell Transport Layer Protocol (August 2009)
- RFC 5656, Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer (December 2009)
- RFC 6187: X.509v3 Certificates for Secure Shell Authentication (March 2011)
- RFC 6239: Suite B Cryptographic Suites for Secure Shell (SSH) (May 2011)
- RFC 6594: Use of the SHA-256 Algorithm with RSA, Digital Signature Algorithm (DSA), and Elliptic Curve DSA (ECDSA) in SSHFP Resource Records
- RFC 6668, SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol (July 2012)
- RFC 7479: Ed25519 SSHFP Resource Records

In addition, the OpenSSH project includes several vendor protocol specifications/extensions:

- OpenSSH PROTOCOL overview (http://cvsweb.openbsd.org/cgi-bin/cvsweb/~checkout~/src/usr.bin/ssh/PROTOCOL?content-type=text/plain)
- OpenSSH certificate/key overview (http://cvsweb.openbsd.org/cgi-bin/cvsweb/~checkout~/src/usr.bin/ssh/PROTOCOL.certkeys?content-type=text/plain)

See also

- Brute-force attack
- Comparison of SSH clients
- Comparison of SSH servers
- Corkscrew
- Ident
- OpenSSH
- Secure Shell tunneling
- Web-based SSH

References

1. Network Working Group of the IETF, January 2006, RFC 4251, The Secure Shell (SSH) Protocol Architecture
2. Network Working Group of the IETF, January 2006, RFC 4252, The Secure Shell (SSH) Authentication Protocol
3. "OpenSSH for Windows" (https://twitter.com/nocentino/status/996843655112613888). Retrieved 8 October 2018.
4. "SSH Hardens the Secure Shell" (http://www.serverwatch.com/news/print.php/3551081). *Serverwatch.com*. Archived (https://web.archive.org/web/20081223200033/http://www.serverwatch.com/news/print.php/3551081) from the original on 2008-12-23.
5. "Prying Eyes: Inside the NSA's War on Internet Security" (http://www.spiegel.de/international/germany/inside-the-nsa-s-war-on-internet-security-a-1010361.html). Spiegel Online. December 28, 2014. Archived (https://web.archive.org/web/20150124202809/http://www.spiegel.de/international/germany/inside-the-nsa-s-war-on-internet-security-a-1010361.html) from the original on January 24, 2015.

6. "How To Set Up Authorized Keys" (http://wiki.qnap.com/wiki/How_To_Set_Up_Authorized_Keys). Archived (https://web.archive.org/web/20110510111514/http://wiki.qnap.com/wiki/How_To_Set_Up_Authorized_Keys) from the original on 2011-05-10.
7. "Service Name and Transport Protocol Port Number Registry" (http://www.iana.org/assignments/port-numbers). *iana.org*. Archived (https://web.archive.org/web/20010604223215/http://www.iana.org/assignments/port-numbers) from the original on 2001-06-04.
8. "Download PuTTY - a free SSH and telnet client for Windows" (http://www.putty.org). Putty.org. Archived (https://web.archive.org/web/20140527122917/http://www.putty.org/) from the original on 2014-05-27. Retrieved 2014-04-28.
9. "Cygwin Package List" (https://cygwin.com/packages/package_list.html). Retrieved January 5, 2016.
10. "WinSCP home page" (http://winscp.net/eng/index.php). Archived (https://web.archive.org/web/20140217163252/http://winscp.net/eng/index.php) from the original on 2014-02-17.
11. "WinSCP page for PortableApps.com" (http://portableapps.com/apps/internet/winscp_portable). Archived (https://web.archive.org/web/20140216120049/http://portableapps.com/apps/internet/winscp_portable) from the original on 2014-02-16.
12. "PuTTY page for PortableApps.com" (http://portableapps.com/apps/internet/putty_portable). Archived (https://web.archive.org/web/20140216214310/http://portableapps.com/apps/internet/putty_portable) from the original on 2014-02-16.
13. Amies, A; Wu, C F; Wang, G C; Criveti, M (2012). "Networking on the cloud" (http://www.ibm.com/developerworks/cloud/library/cl-networkingtools/index.html). *IBM developerWorks*. Archived (https://web.archive.org/web/20130614123106/http://www.ibm.com/developerworks/cloud/library/cl-networkingtools/index.html) from the original on 2013-06-14.
14. Tatu Ylönen. "The new skeleton key: changing the locks in your network environment" (https://www.scmagazineuk.com/the-new-skeleton-key-changing-the-locks-in-your-network-environment/article/545848/). Archived (https://web.archive.org/web/20170820162632/https://www.scmagazineuk.com/the-new-skeleton-key-changing-the-locks-in-your-network-environment/article/545848/) from the original on 2017-08-20.
15. Tatu Ylönen. "SSH Port" (https://www.ssh.com/ssh/port). Archived (https://web.archive.org/web/20170803235736/https://www.ssh.com/ssh/port) from the original on 2017-08-03.
16. Nicholas Rosasco and David Larochelle. "How and Why More Secure Technologies Succeed in Legacy Markets: Lessons from the Success of SSH" (http://www.cs.virginia.edu/~drl7x/sshVsTelnetWeb3.pdf) (PDF). *Quoting* ***Barrett*** *and Silverman,* SSH, the Secure Shell: The Definitive Guide, *O'Reilly & Associates (2001)*. Dept. of Computer Science, Univ. of Virginia. Archived (https://web.archive.org/web/20060625065258/http://www.cs.virginia.edu/~drl7x/sshVsTelnetWeb3.pdf) (PDF) from the original on 2006-06-25. Retrieved 2006-05-19.
17. "Secsh Protocol Documents" (http://www.vandyke.com/technology/drafts.html). *VanDyke Software, Inc*. Archived (https://web.archive.org/web/20100113104155/http://vandyke.com/technology/drafts.html) from the original on 2010-01-13.
18. "SSH Frequently Asked Questions" (http://www.snailbook.com/faq/ssh-1-vs-2.auto.html). Archived (https://web.archive.org/web/20041010035705/http://www.snailbook.com/faq/ssh-1-vs-2.auto.html) from the original on 2004-10-10.
19. "libssh" (https://www.libssh.org/2018/08/10/libssh-0-8-0/).
20. "A GNU implementation of the Secure Shell protocols" (http://www.lysator.liu.se/~nisse/lsh/). Archived (https://web.archive.org/web/20120204035753/http://www.lysator.liu.se/~nisse/lsh/) from the original on 2012-02-04.
21. "Dropbear SSH" (https://matt.ucc.asn.au/dropbear/dropbear.html). Archived (https://web.archive.org/web/20111014091250/http://matt.ucc.asn.au/dropbear/dropbear.html) from the original on 2011-10-14.
22. "RFC 4253" (http://tools.ietf.org/html/rfc4253#section-5.1). Section 5. Compatibility With Old SSH Versions. Archived (https://web.archive.org/web/20100704150937/http://tools.ietf.org/html/rfc4253#section-5.1) from the original on 2010-07-04., IETF
23. "OpenSSH: Project History and Credits" (http://www.openssh.com/history.html). openssh.com. 2004-12-22. Archived (https://web.archive.org/web/20131224105341/http://openssh.com/history.html) from the original on 2013-12-24. Retrieved 2014-04-27.

24. "OSSH Information for VU#419241" (https://www.kb.cert.org/vuls/id/MIMG-6L4LBL). Archived (https://web.archive.org/web/20070927231942/https://www.kb.cert.org/vuls/id/MIMG-6L4LBL) from the original on 2007-09-27.
25. Sobell, Mark (2012). *A Practical Guide to Linux Commands, Editors, and Shell Programming (3rd Edition)*. Upper Saddle River, NJ: Prentice Hall. pp. 702–704. ISBN 978-0133085044.
26. Seggelmann, R.; Tuxen, M.; Rathgeb, E.P. (18–20 July 2012). "SSH over SCTP — Optimizing a multi-channel protocol by adapting it to SCTP". *Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2012 8th International Symposium on*: 1–6. doi:10.1109/CSNDSP.2012.6292659 (https://doi.org/10.1109%2FCSNDSP.2012.6292659). ISBN 978-1-4577-1473-3.
27. Stebila, D.; Green J. (December 2009). "RFC5656 - Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer" (https://tools.ietf.org/html/rfc5656). Archived (https://web.archive.org/web/20120719131525/http://tools.ietf.org/html/rfc5656) from the original on 19 July 2012. Retrieved 12 November 2012.
28. Miller, D.; Valchev, P. (September 3, 2007). "The use of UMAC in the SSH Transport Layer Protocol / draft-miller-secsh-umac-00.txt" (https://tools.ietf.org/html/draft-miller-secsh-umac-01). Archived (https://web.archive.org/web/20140819125608/https://tools.ietf.org/html/draft-miller-secsh-umac-01) from the original on 19 August 2014. Retrieved 12 November 2012.
29. "SSH Insertion Attack" (http://www.coresecurity.com/content/ssh-insertion-attack). *Core Security Technologies*. Archived (https://web.archive.org/web/20110708192336/http://www.coresecurity.com/content/ssh-insertion-attack) from the original on 2011-07-08.
30. "Vulnerability Note VU#13877 - Weak CRC allows packet injection into SSH sessions encrypted with block ciphers" (http://www.kb.cert.org/vuls/id/13877). *US CERT*. Archived (https://web.archive.org/web/20100710040357/http://www.kb.cert.org/vuls/id/13877) from the original on 2010-07-10.
31. "SSH CRC-32 Compensation Attack Detector Vulnerability" (http://www.securityfocus.com/bid/2347/discuss). *SecurityFocus*. Archived (https://web.archive.org/web/20080725110345/http://www.securityfocus.com/bid/2347/discuss) from the original on 2008-07-25.
32. "Vulnerability Note VU#945216 - SSH CRC32 attack detection code contains remote integer overflow" (http://www.kb.cert.org/vuls/id/945216). *US CERT*. Archived (https://web.archive.org/web/20051013074750/http://www.kb.cert.org/vuls/id/945216) from the original on 2005-10-13.
33. "Vulnerability Note VU#315308 - Weak CRC allows last block of IDEA-encrypted SSH packet to be changed without notice" (http://www.kb.cert.org/vuls/id/315308). *US CERT*. Archived (https://web.archive.org/web/20100711103528/http://www.kb.cert.org/vuls/id/315308) from the original on 2010-07-11.
34. "Vulnerability Note VU#684820 - SSH-1 allows client authentication to be forwarded by a malicious server to another server" (http://www.kb.cert.org/vuls/id/684820). *US CERT*. Archived (https://web.archive.org/web/20090901012536/http://www.kb.cert.org/vuls/id/684820) from the original on 2009-09-01.
35. "Vulnerability Note VU#958563 - SSH CBC vulnerability" (http://www.kb.cert.org/vuls/id/958563). *US CERT*. Archived (https://web.archive.org/web/20110622005639/http://www.kb.cert.org/vuls/id/958563) from the original on 2011-06-22.
36. Tatu Ylonen. "BothanSpy & Gyrfalcon - Analysis of CIA hacking tools for SSH (https://www.ssh.com/ssh/cia-bothanspy-gyrfalcon)", ssh.com, 3 August 2017. Retrieved 15 july 2018.

Further reading

- Daniel J. Barrett, Richard E. Silverman, and Robert G. Byrnes, *SSH: The Secure Shell (The Definitive Guide)*, O'Reilly 2005 (2nd edition). ISBN 0-596-00895-3
- Michael Stahnke, *Pro OpenSSH*, Apress 2005 ISBN 1-59059-476-2
- Tatu Ylönen (12 July 1995). "Announcement: Ssh (Secure Shell) Remote Login Program" (https://groups.google.com/group/comp.security.unix/msg/67079d812a19f499?dmode=source&hl=en). comp.security.unix. Original announcement of Ssh
- Himanshu Dwivedi; *Implementing SSH*, Wiley 2003. ISBN 978-0-471-45880-7
- This article is based on material taken from the *Free On-line Dictionary of Computing* prior to 1 November 2008 and incorporated under the "relicensing" terms of the GFDL, version 1.3 or later.

External links

- SSH Protocols (http://www.snailbook.com/protocols.html)
- RFC7076
- How to establish passwordless login with ssh (http://vfx.engineering/2014/05/11/infrastructure-how-to-establish-passwordless-login-with-ssh/)

WIKIPEDIA

Simple Mail Transfer Protocol

The **Simple Mail Transfer Protocol** (**SMTP**) is a communication protocol for electronic mail transmission. As an Internet standard, SMTP was first defined in 1982 by RFC 821 (https://tools.ietf.org/html/rfc821), and updated in 2008 by RFC 5321 (https://tools.ietf.org/html/rfc5321) to Extended SMTP additions, which is the protocol variety in widespread use today. Mail servers and other message transfer agents use SMTP to send and receive mail messages. Proprietary systems such as Microsoft Exchange and IBM Notes and webmail systems such as Outlook.com, Gmail and Yahoo! Mail may use non-standard protocols internally, but all use SMTP when sending to or receiving email from outside their own systems. SMTP servers commonly use the Transmission Control Protocol on port number 25.

User-level email clients typically use SMTP only for sending messages to a mail server for relaying, typically submit outgoing email to the mail server on port 587 or 465 as per RFC 8314. For retrieving messages, IMAP and POP3 are standard, but proprietary servers also often implement proprietary protocols, e.g., Exchange ActiveSync.

Contents

History

Various forms of one-to-one electronic messaging were used in the 1960s. Users communicated using systems developed for specific mainframe computers. As more computers were interconnected, especially in the U.S. Government's ARPANET, standards were developed to permit exchange of messages between different operating

systems. SMTP grew out of these standards developed during the 1970s.

SMTP traces its roots to two implementations described in 1971: the Mail Box Protocol, whose implementation has been disputed,[1] but is discussed in RFC 196 (https://tools.ietf.org/html/rfc196) and other RFCs, and the SNDMSG program, which, according to RFC 2235 (https://tools.ietf.org/html/rfc2235), Ray Tomlinson of BBN invented for TENEX computers to send mail messages across the ARPANET.[2][3][4] Fewer than 50 hosts were connected to the ARPANET at this time.[5]

Further implementations include FTP Mail[6] and Mail Protocol, both from 1973.[7] Development work continued throughout the 1970s, until the ARPANET transitioned into the modern Internet around 1980. Jon Postel then proposed a Mail Transfer Protocol in 1980 that began to remove the mail's reliance on FTP.[8] SMTP was published as RFC 788 (https://tools.ietf.org/html/rfc788) in November 1981, also by Postel.

The SMTP standard was developed around the same time as Usenet, a one-to-many communication network with some similarities.

SMTP became widely used in the early 1980s. At the time, it was a complement to Unix to Unix Copy Program (UUCP) mail, which was better suited for handling email transfers between machines that were intermittently connected. SMTP, on the other hand, works best when both the sending and receiving machines are connected to the network all the time. Both use a store and forward mechanism and are examples of push technology. Though Usenet's newsgroups are still propagated with UUCP between servers,[9] UUCP as a mail transport has virtually disappeared[10] along with the "bang paths" it used as message routing headers.[11]

Sendmail, released with 4.1cBSD, right after RFC 788 (https://tools.ietf.org/html/rfc788), was one of the first mail transfer agents to implement SMTP.[12] Over time, as BSD Unix became the most popular operating system on the Internet, sendmail became the most common MTA (mail transfer agent).[13] Some other popular SMTP server programs include Postfix, qmail, Novell GroupWise, Exim, Novell NetMail, Microsoft Exchange Server and Oracle Communications Messaging Server.

Message submission (RFC 2476 (https://tools.ietf.org/html/rfc2476)) and SMTP-AUTH (RFC 2554 (https://tools.ietf.org/html/rfc2554)) were introduced in 1998 and 1999, both describing new trends in email delivery. Originally, SMTP servers were typically internal to an organization, receiving mail for the organization *from the outside*, and relaying messages from the organization *to the outside*. But as time went on, SMTP servers (mail transfer agents), in practice, were expanding their roles to become message submission agents for Mail user agents, some of which were now relaying mail *from the outside* of an organization. (e.g. a company executive wishes to send email while on a trip using the corporate SMTP server.) This issue, a consequence of the rapid expansion and popularity of the World Wide Web, meant that SMTP had to include specific rules and methods for relaying mail and authenticating users to prevent abuses such as relaying of unsolicited email (spam). Work on message submission (RFC 2476 (https://tools.ietf.org/html/rfc2476)) was originally started because popular mail servers would often rewrite mail in an attempt to fix problems in it, for example, adding a domain name to an unqualified address. This behavior is helpful when the message being fixed is an initial submission, but dangerous and harmful when the message originated elsewhere and is being relayed. Cleanly separating mail into submission and relay was seen as a way to permit and encourage rewriting submissions while prohibiting rewriting relay. As spam became more prevalent, it was also seen as a way to provide authorization for mail being sent out from an organization, as well as traceability. This separation of relay and submission quickly became a foundation for modern email security practices.

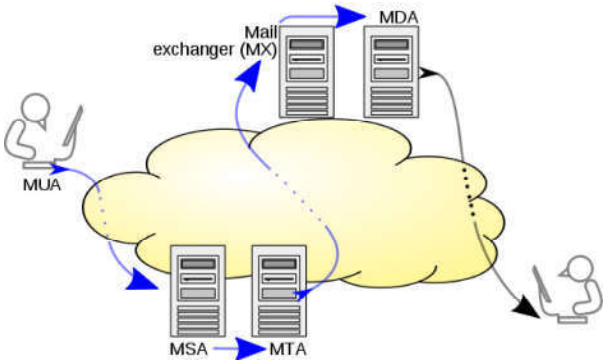
As this protocol started out purely ASCII text-based, it did not deal well with binary files, or characters in many non-English languages. Standards such as Multipurpose Internet Mail Extensions (MIME) were developed to encode binary files for transfer through SMTP. Mail transfer agents (MTAs) developed after Sendmail also tended to be implemented 8-bit-clean, so that the alternate "just send eight" strategy could be used to transmit arbitrary text data (in any 8-bit ASCII-like character encoding) via SMTP. Mojibake was still a problem due to differing character set mappings between vendors, although the email addresses themselves still allowed only ASCII. 8-bit-clean MTAs today

tend to support the 8BITMIME extension, permitting binary files to be transmitted almost as easily as plain text. Recently the SMTPUTF8 extension was created to support UTF-8 text, allowing international content and addresses in non-Latin scripts like Cyrillic or Chinese.

Many people contributed to the core SMTP specifications, among them Jon Postel, Eric Allman, Dave Crocker, Ned Freed, Randall Gellens, John Klensin, and Keith Moore.

Mail processing model

Email is submitted by a mail client (mail user agent, MUA) to a mail server (mail submission agent, MSA) using SMTP on TCP port 587. Most mailbox providers still allow submission on traditional port 25. The MSA delivers the mail to its mail transfer agent (mail transfer agent, MTA). Often, these two agents are instances of the same software launched with different options on the same machine. Local processing can be done either on a single machine, or split among multiple machines; mail agent processes on one machine can share files, but if processing is on multiple machines, they transfer messages between each other using SMTP, where each machine is configured to use the next machine as a smart host. Each process is an MTA (an SMTP server) in its own right.



Blue arrows depict implementation of SMTP variations.

The boundary MTA uses the Domain name system (DNS) to look up the mail exchanger record (MX record) for the recipient's domain (the part of the email address on the right of @). The MX record contains the name of the target host. Based on the target host and other factors, the MTA selects an exchange server: see the article MX record. The MTA connects to the exchange server as an SMTP client.

Message transfer can occur in a single connection between two MTAs, or in a series of hops through intermediary systems. A receiving SMTP server may be the ultimate destination, an intermediate "relay" (that is, it stores and forwards the message) or a "gateway" (that is, it may forward the message using some protocol other than SMTP). Each hop is a formal handoff of responsibility for the message, whereby the receiving server must either deliver the message or properly report the failure to do so.[14]

Once the final hop accepts the incoming message, it hands it to a mail delivery agent (MDA) for local delivery. An MDA saves messages in the relevant mailbox format. As with sending, this reception can be done using one or multiple computers, but in the diagram above the MDA is depicted as one box near the mail exchanger box. An MDA may deliver messages directly to storage, or forward them over a network using SMTP or other protocol such as Local Mail Transfer Protocol (LMTP), a derivative of SMTP designed for this purpose.

Once delivered to the local mail server, the mail is stored for batch retrieval by authenticated mail clients (MUAs). Mail is retrieved by end-user applications, called email clients, using Internet Message Access Protocol (IMAP), a protocol that both facilitates access to mail and manages stored mail, or the Post Office Protocol (POP) which typically uses the traditional mbox mail file format or a proprietary system such as Microsoft Exchange/Outlook or Lotus Notes/Domino. Webmail clients may use either method, but the retrieval protocol is often not a formal standard.

SMTP defines message *transport*, not the message *content*. Thus, it defines the mail *envelope* and its parameters, such as the envelope sender, but not the header (except *trace information*) nor the body of the message itself. STD 10 and RFC 5321 (https://tools.ietf.org/html/rfc5321) define SMTP (the envelope), while STD 11 and RFC 5322 (https://tools.ietf.org/html/rfc5322) define the message (header and body), formally referred to as the Internet Message Format.

Protocol overview

SMTP is a connection-oriented, text-based protocol in which a mail sender communicates with a mail receiver by issuing command strings and supplying necessary data over a reliable ordered data stream channel, typically a Transmission Control Protocol (TCP) connection. An *SMTP session* consists of commands originated by an SMTP client (the initiating agent, sender, or transmitter) and corresponding responses from the SMTP server (the listening agent, or receiver) so that the session is opened, and session parameters are exchanged. A session may include zero or more SMTP transactions. An *SMTP transaction* consists of three command/reply sequences:

1. **MAIL** command, to establish the return address, also called return-path,[15] reverse-path,[16] bounce address, mfrom, or envelope sender.
2. **RCPT** command, to establish a recipient of the message. This command can be issued multiple times, one for each recipient. These addresses are also part of the envelope.
3. **DATA** to signal the beginning of the *message text*; the content of the message, as opposed to its envelope. It consists of a *message header* and a *message body* separated by an empty line. DATA is actually a group of commands, and the server replies twice: once to the *DATA command* itself, to acknowledge that it is ready to receive the text, and the second time after the end-of-data sequence, to either accept or reject the entire message.

Besides the intermediate reply for DATA, each server's reply can be either positive (2xx reply codes) or negative. Negative replies can be permanent (5xx codes) or transient (4xx codes). A **reject** is a permanent failure and the client should send a bounce message to the server it received it from. A **drop** is a positive response followed by message discard rather than delivery.

The initiating host, the SMTP client, can be either an end-user's email client, functionally identified as a mail user agent (MUA), or a relay server's mail transfer agent (MTA), that is an SMTP server acting as an SMTP client, in the relevant session, in order to relay mail. Fully capable SMTP servers maintain queues of messages for retrying message transmissions that resulted in transient failures.

A MUA knows the *outgoing mail* SMTP server from its configuration. A relay server typically determines which server to connect to by looking up the MX (Mail eXchange) DNS resource record for each recipient's domain name. If no MX record is found, a conformant relaying server (not all are) instead looks up the A record. Relay servers can also be configured to use a smart host. A relay server initiates a TCP connection to the server on the "well-known port" for SMTP: port 25, or for connecting to an MSA, port 587. The main difference between an MTA and an MSA is that connecting to an MSA requires SMTP Authentication.

SMTP vs mail retrieval

SMTP is a delivery protocol only. In normal use, mail is "pushed" to a destination mail server (or next-hop mail server) as it arrives. Mail is routed based on the destination server, not the individual user(s) to which it is addressed. Other protocols, such as the Post Office Protocol (POP) and the Internet Message Access Protocol (IMAP) are specifically designed for use by individual users retrieving messages and managing mail boxes. To permit an intermittently-connected mail server to *pull* messages from a remote server on demand, SMTP has a feature to initiate mail queue processing on a remote server (see Remote Message Queue Starting below). POP and IMAP are unsuitable protocols for relaying mail by intermittently-connected machines; they are designed to operate after final delivery, when information critical to the correct operation of mail relay (the "mail envelope") has been removed.

Remote Message Queue Starting

Remote Message Queue Starting is a feature of SMTP that permits a remote host to start processing of the mail queue on a server so it may receive messages destined to it by sending the `TURN` command. This feature however was deemed insecure[17] and was extended in RFC 1985 (https://tools.ietf.org/html/rfc1985) with the `ETRN` command which operates more securely using an authentication method based on Domain Name System information.

On-Demand Mail Relay

On-Demand Mail Relay (**ODMR**) is an SMTP extension standardized in RFC 2645 (https://tools.ietf.org/html/rfc2645) that allows an intermittently-connected SMTP server to receive email queued for it when it is connected.

Internationalization

Users whose native script is not Latin based, or who use diacritic not in the ASCII character set have had difficulty with the Latin email address requirement. RFC 6531 (https://tools.ietf.org/html/rfc6531) was created to solve that problem, providing internationalization features for SMTP, the SMTPUTF8 extension and support for multi-byte and non-ASCII characters in email addresses, such as those with diacritics and other language characters such as Greek and Chinese.[18]

Current support is limited, but there is strong interest in broad adoption of RFC 6531 (https://tools.ietf.org/html/rfc6531) and the related RFCs in countries like China that have a large user base where Latin (ASCII) is a foreign script.

Outgoing mail SMTP server

An email client needs to know the IP address of its initial SMTP server and this has to be given as part of its configuration (usually given as a DNS name). This server will deliver outgoing messages on behalf of the user.

Outgoing mail server access restrictions

Server administrators need to impose some control on which clients can use the server. This enables them to deal with abuse, for example spam. Two solutions have been in common use:

- In the past, many systems imposed usage restrictions by the *location* of the client, only permitting usage by clients whose IP address is one that the server administrators control. Usage from any other client IP address is disallowed.
- Modern SMTP servers typically offer an alternative system that requires authentication of clients by credentials before allowing access.

Restricting access by location

Under this system, an ISP's SMTP server will not allow access by users who are outside the ISP's network. More precisely, the server may only allow access to users with an IP address provided by the ISP, which is equivalent to requiring that they are connected to the Internet using that same ISP. A mobile user may often be on a network other than that of their normal ISP, and will then find that sending email fails because the configured SMTP server choice is no longer accessible.

This system has several variations. For example, an organisation's SMTP server may only provide service to users on the same network, enforcing this by firewalling to block access by users on the wider Internet. Or the server may perform range checks on the client's IP address. These methods were typically used by corporations and institutions such as universities which provided an SMTP server for outbound mail only for use internally within the organisation. However, most of these bodies now use client authentication methods, as described below.

Where a user is mobile, and may use different ISPs to connect to the internet, this kind of usage restriction is onerous, and altering the configured outbound email SMTP server address is impractical. It is highly desirable to be able to use email client configuration information that does not need to change.

Client authentication

Modern SMTP servers typically require authentication of clients by credentials before allowing access, rather than restricting access by location as described earlier. This more flexible system is friendly to mobile users and allows them to have a fixed choice of configured outbound SMTP server. SMTP Authentication, often abbreviated SMTP AUTH, is an extension of the SMTP in order to log in using an authentication mechanism.

Open relay

A server that is accessible on the wider Internet and does not enforce these kinds of access restrictions is known as an open relay. This is now generally considered a bad practice worthy of blacklisting.

Ports

Communication between mail servers generally uses the standard TCP port 25 designated for SMTP.

Mail *clients* however generally don't use this, instead using specific "submission" ports. Mail services generally accept email submission from clients on one of:

- 587 (Submission), as formalized in RFC 6409 (https://tools.ietf.org/html/rfc6409) (previously RFC 2476 (https://tools.ietf.org/html/rfc2476))
- 465 This port was deprecated after RFC 2487 (https://tools.ietf.org/html/rfc2487), until the issue of RFC 8314.

Port 2525 and others may be used by some individual providers, but have never been officially supported.

Most Internet service providers now block all outgoing port 25 traffic from their customers as an anti-spam measure.[19] For the same reason, businesses will typically configure their firewall to only allow outgoing port 25 traffic from their designated mail servers.

SMTP transport example

A typical example of sending a message via SMTP to two mailboxes (*alice* and *theboss*) located in the same mail domain (*example.com* or *localhost.com*) is reproduced in the following session exchange. (In this example, the conversation parts are prefixed with *S:* and *C:*, for *server* and *client*, respectively; these labels are not part of the exchange.)

After the message sender (SMTP client) establishes a reliable communications channel to the message receiver (SMTP server), the session is opened with a greeting by the server, usually containing its fully qualified domain name (FQDN), in this case *smtp.example.com*. The client initiates its dialog by responding with a `HELO` command identifying itself in the command's parameter with its FQDN (or an address literal if none is available).[20]

```
S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.com
S: 250 smtp.example.com, I am glad to meet you
C: MAIL FROM:<bob@example.com>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: From: "Bob Example" <bob@example.com>
C: To: Alice Example <alice@example.com>
C: Cc: theboss@example.com
C: Date: Tue, 15 Jan 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
```

```
C: This is a test message with 5 header fields and 4 lines in the message body.
C: Your friend,
C: Bob
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye
{The server closes the connection}
```

The client notifies the receiver of the originating email address of the message in a `MAIL FROM` command. This is also the return or bounce address in case the message cannot be delivered. In this example the email message is sent to two mailboxes on the same SMTP server: one for each recipient listed in the `To` and `Cc` header fields. The corresponding SMTP command is `RCPT TO`. Each successful reception and execution of a command is acknowledged by the server with a result code and response message (e.g., 250 Ok).

The transmission of the body of the mail message is initiated with a `DATA` command after which it is transmitted verbatim line by line and is terminated with an end-of-data sequence. This sequence consists of a new-line (<CR> <LF>), a single full stop (period), followed by another new-line. Since a message body can contain a line with just a period as part of the text, the client sends *two* periods every time a line starts with a period; correspondingly, the server replaces every sequence of two periods at the beginning of a line with a single one. Such escaping method is called *dot-stuffing*.

The server's positive reply to the end-of-data, as exemplified, implies that the server has taken the responsibility of delivering the message. A message can be doubled if there is a communication failure at this time, e.g. due to a power shortage: Until the sender has received that `250` reply, it must assume the message was not delivered. On the other hand, after the receiver has decided to accept the message, it must assume the message has been delivered to it. Thus, during this time span, both agents have active copies of the message that they will try to deliver.[21] The probability that a communication failure occurs exactly at this step is directly proportional to the amount of filtering that the server performs on the message body, most often for anti-spam purposes. The limiting timeout is specified to be 10 minutes.[22]

The `QUIT` command ends the session. If the email has other recipients located elsewhere, the client would `QUIT` and connect to an appropriate SMTP server for subsequent recipients after the current destination(s) had been queued. The information that the client sends in the `HELO` and `MAIL FROM` commands are added (not seen in example code) as additional header fields to the message by the receiving server. It adds a `Received` and `Return-Path` header field, respectively.

Some clients are implemented to close the connection after the message is accepted (`250 Ok: queued as 12345`), so the last two lines may actually be omitted. This causes an error on the server when trying to send the `221` reply.

Optional extensions

Clients learn a server's supported options by using the `EHLO` greeting, as exemplified below, instead of the original `HELO` (example above). Clients fall back to `HELO` only if the server does not support SMTP extensions.[23]

Modern clients may use the ESMTP extension keyword `SIZE` to query the server for the maximum message size that will be accepted. Older clients and servers may try to transfer excessively sized messages that will be rejected after consuming network resources, including connect time to network links that is paid by the minute.[24]

Users can manually determine in advance the maximum size accepted by ESMTP servers. The client replaces the `HELO` command with the `EHLO` command.

```
S: 220 smtp2.example.com ESMTP Postfix
C: EHLO bob.example.com
S: 250-smtp2.example.com Hello bob.example.org [192.0.2.201]
```

```
S: 250-SIZE 14680064
S: 250-PIPELINING
S: 250 HELP
```

Thus *smtp2.example.com* declares that can accept a fixed maximum message size no larger than 14,680,064 octets (8-bit bytes).

In the simplest case, an ESMTP server declares a maximum `SIZE` immediately after receiving an `EHLO`. According to RFC 1870 (https://tools.ietf.org/html/rfc1870), however, the numeric parameter to the `SIZE` extension in the `EHLO` response is optional. Clients may instead, when issuing a `MAIL FROM` command, include a numeric estimate of the size of the message they are transferring, so that the server can refuse receipt of overly-large messages.

Spoofing and spamming

The original design of SMTP had no facility to authenticate senders, or check that servers were authorized to send on their behalf, with the result that email spoofing is possible, and commonly used in email spam and phishing.

Occasional proposals are made to modify SMTP extensively or replace it completely. One example of this is Internet Mail 2000, but neither it, nor any other has made much headway in the face of the network effect of the huge installed base of classic SMTP. Instead, mail servers now use a range of techniques, including DomainKeys Identified Mail, Sender Policy Framework and DMARC, DNSBLs and greylisting to reject or quarantine suspicious emails.

Implementations

Related requests for comments

- RFC 1123 (https://tools.ietf.org/html/rfc1123) – Requirements for Internet Hosts—Application and Support (STD 3)
- RFC 1870 (https://tools.ietf.org/html/rfc1870) – SMTP Service Extension for Message Size Declaration (obsoletes: RFC 1653 (https://tools.ietf.org/html/rfc1653))
- RFC 2505 (https://tools.ietf.org/html/rfc2505) – Anti-Spam Recommendations for SMTP MTAs (BCP 30)
- RFC 2920 (https://tools.ietf.org/html/rfc2920) – SMTP Service Extension for Command Pipelining (STD 60)
- RFC 3030 (https://tools.ietf.org/html/rfc3030) – SMTP Service Extensions for Transmission of Large and Binary MIME Messages
- RFC 3207 (https://tools.ietf.org/html/rfc3207) – SMTP Service Extension for Secure SMTP over Transport Layer Security (obsoletes RFC 2487 (https://tools.ietf.org/html/rfc2487))
- RFC 3461 (https://tools.ietf.org/html/rfc3461) – SMTP Service Extension for Delivery Status Notifications (obsoletes RFC 1891 (https://tools.ietf.org/html/rfc1891))
- RFC 3463 (https://tools.ietf.org/html/rfc3463) – Enhanced Status Codes for SMTP (obsoletes RFC 1893 (https://tools.ietf.org/html/rfc1893), updated by RFC 5248 (https://tools.ietf.org/html/rfc5248))
- RFC 3464 (https://tools.ietf.org/html/rfc3464) – An Extensible Message Format for Delivery Status Notifications (obsoletes RFC 1894 (https://tools.ietf.org/html/rfc1894))
- RFC 3798 (https://tools.ietf.org/html/rfc3798) – Message Disposition Notification (updates RFC 3461 (https://tools.ietf.org/html/rfc3461))
- RFC 3834 (https://tools.ietf.org/html/rfc3834) – Recommendations for Automatic Responses to Electronic Mail
- RFC 4952 (https://tools.ietf.org/html/rfc4952) – Overview and Framework for Internationalized Email (updated by RFC 5336 (https://tools.ietf.org/html/rfc5336))
- RFC 4954 (https://tools.ietf.org/html/rfc4954) – SMTP Service Extension for Authentication (obsoletes RFC 2554 (https://tools.ietf.org/html/rfc2554), updates RFC 3463 (https://tools.ietf.org/html/rfc3463), updated by RFC 5248 (https://tools.ietf.org/html/rfc5248))

- RFC 5068 (https://tools.ietf.org/html/rfc5068) – Email Submission Operations: Access and Accountability Requirements (BCP 134)
- RFC 5248 (https://tools.ietf.org/html/rfc5248) – A Registry for SMTP Enhanced Mail System Status Codes (BCP 138) (updates RFC 3463 (https://tools.ietf.org/html/rfc3463))
- RFC 5321 (https://tools.ietf.org/html/rfc5321) – The Simple Mail Transfer Protocol (obsoletes RFC 821 (https://tools.ietf.org/html/rfc821) aka STD 10, RFC 974 (https://tools.ietf.org/html/rfc974), RFC 1869 (https://tools.ietf.org/html/rfc1869), RFC 2821 (https://tools.ietf.org/html/rfc2821), updates RFC 1123 (https://tools.ietf.org/html/rfc1123))
- RFC 5322 (https://tools.ietf.org/html/rfc5322) – Internet Message Format (obsoletes RFC 822 (https://tools.ietf.org/html/rfc822) aka STD 11, and RFC 2822 (https://tools.ietf.org/html/rfc2822))
- RFC 5504 (https://tools.ietf.org/html/rfc5504) – Downgrading Mechanism for Email Address Internationalization
- RFC 6409 (https://tools.ietf.org/html/rfc6409) – Message Submission for Mail (STD 72) (obsoletes RFC 4409 (https://tools.ietf.org/html/rfc4409), RFC 2476 (https://tools.ietf.org/html/rfc2476))
- RFC 6522 (https://tools.ietf.org/html/rfc6522) – The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages (obsoletes RFC 3462 (https://tools.ietf.org/html/rfc3462), and in turn RFC 1892 (https://tools.ietf.org/html/rfc1892))
- RFC 6531 (https://tools.ietf.org/html/rfc6531) – SMTP Extension for Internationalized Email Addresses (updates RFC 2821 (https://tools.ietf.org/html/rfc2821), RFC 2822 (https://tools.ietf.org/html/rfc2822), RFC 4952 (https://tools.ietf.org/html/rfc4952), and RFC 5336 (https://tools.ietf.org/html/rfc5336))
- RFC 8314 - Cleartext Considered Obsolete: Use of Transport Layer Security (TLS) for Email Submission and Access

See also

- Bounce address
- Email encryption
- Ident
- POP before SMTP / SMTP after POP
- Sender Policy Framework (SPF)
- Variable envelope return path
- DKIM

Notes

1. The History of Electronic Mail (http://www.multicians.org/thvv/mail-history.html), *Tom Van Vleck*: "It is not clear this protocol was ever implemented "
2. *The First Network Email* (https://openmap.bbn.com/~tomlinso/ray/firstemailframe.html), Ray Tomlinson, BBN
3. Picture of "The First Email Computer (https://openmap.bbn.com/~tomlinso/ray/ka10.html)" by Dan Murphy, a PDP-10
4. Dan Murphy's TENEX and TOPS-20 Papers (http://www.opost.com/dlm/tenex/) Archived (https://web.archive.org/web/20071118204016/http://www.opost.com/dlm/tenex/) November 18, 2007, at the Wayback Machine
5. RFC 2235 (https://tools.ietf.org/html/rfc2235)
6. RFC 469 (https://tools.ietf.org/html/rfc469) – Network Mail Meeting Summary
7. RFC 524 (https://tools.ietf.org/html/rfc524) – A Proposed Mail Protocol
8. RFC 772 (https://tools.ietf.org/html/rfc772) – Mail Transfer Protocol
9. Tldp.org (http://tldp.org/HOWTO/Usenet-News-HOWTO/x64.html)
10. draft-barber-uucp-project-conclusion-05 – The Conclusion of the UUCP Mapping Project (https://tools.ietf.org/html/draft-barber-uucp-project-conclusion-05)

11. The article about sender rewriting contains technical background info about the early SMTP history and source routing before RFC 1123 (https://tools.ietf.org/html/rfc1123).
12. Eric Allman (1983), *Sendmail – An Internetwork Mail Router* (https://docs.freebsd.org/44doc/smm/09.sendmail/paper.pdf) (PDF), BSD UNIX documentation set, Berkeley: University of California, retrieved June 29, 2012
13. Craig Partridge (2008), *The Technical Development of Internet Email* (https://web.archive.org/web/20110512165437/http://www.ir.bbn.com/~craig/email.pdf) (PDF), IEEE Annals of the History of Computing, **30** (2), IEEE Computer Society, pp. 3–29, doi:10.1109/MAHC.2008.32 (https://doi.org/10.1109%2FMAHC.2008.32), archived from the original (http://www.ir.bbn.com/~craig/email.pdf) (PDF) on May 12, 2011
14. John Klensin (October 2008). "Basic Structure" (https://tools.ietf.org/html/rfc5321#section-2.1). *Simple Mail Transfer Protocol* (https://tools.ietf.org/html/rfc5321). IETF. sec. 2.1. doi:10.17487/RFC5321 (https://doi.org/10.17487%2FRFC5321). RFC 5321. Retrieved 16 January 2016.
15. "The MAIL, RCPT, and DATA verbs" (http://cr.yp.to/smtp/mail.html), [D. J. Bernstein]
16. RFC 5321 (https://tools.ietf.org/html/rfc5321) Section-7.2
17. RFC 1985 (https://tools.ietf.org/html/rfc1985), *SMTP Service Extension for Remote Message Queue Starting*, J. De Winter, The Internet Society (August 1996)
18. Jiankang Yao (19 December 2014). "Chinese email address" (http://www.ietf.org/mail-archive/web/ima/current/msg05395.html). *EAI* (Mailing list). IETF. Retrieved 24 May 2016.
19. Cara Garretson (2005). "ISPs Pitch In to Stop Spam" (http://www.pcworld.com/article/116843/article.html). PC World. Retrieved 18 January 2016. "Last month, the Anti-Spam Technical Alliance, formed last year by Yahoo, America Online, EarthLink, and Microsoft, issued a list of antispam recommendations that includes filtering Port 25."
20. RFC 5321 (https://tools.ietf.org/html/rfc5321), *Simple Mail Transfer Protocol*, J. Klensin, The Internet Society (October 2008)
21. RFC 1047 (https://tools.ietf.org/html/rfc1047)
22. rfc5321#section-4.5.3.2.6 (https://tools.ietf.org/html/rfc5321#section-4.5.3.2.6)
23. John Klensin; Ned Freed; Marshall T. Rose; Einar A. Stefferud; Dave Crocker (November 1995). *SMTP Service Extensions* (https://tools.ietf.org/html/rfc1869). IETF. doi:10.17487/RFC1869 (https://doi.org/10.17487%2FRFC1869). RFC 1869.
24. "MAIL Parameters" (http://www.iana.org/assignments/mail-parameters/mail-parameters.txt). IANA. Retrieved 3 April 2016.

References

- Hughes, L (1998). *Internet E-mail: Protocols, Standards and Implementation*. Artech House Publishers. ISBN 978-0-89006-939-4.
- Hunt, C (2003). *sendmail Cookbook*. O'Reilly Media. ISBN 978-0-596-00471-2.
- Johnson, K (2000). *Internet Email Protocols: A Developer's Guide*. Addison-Wesley Professional. ISBN 978-0-201-43288-6.
- Loshin, P (1999). *Essential Email Standards: RFCs and Protocols Made Practical*. John Wiley & Sons. ISBN 978-0-471-34597-8.
- Rhoton, J (1999). *Programmer's Guide to Internet Mail: SMTP, POP, IMAP, and LDAP*. Elsevier. ISBN 978-1-55558-212-8.
- Wood, D (1999). *Programming Internet Mail*. O'Reilly. ISBN 978-1-56592-479-6.